

## 5. CONCLUSIONS AND RECOMMENDATIONS

From the validation study and sensitivity study, we concluded that MADCAP is a reasonable regional photochemical model to study the air quality of the San Diego Air Basin.

In comparison with the measured data, MADCAP predictions show a remarkable agreement in both transport and ozone generation with a one hour time shift (allowable in the solution method). The 10/3/75 validation run shows a better agreement with data than the 10/16/75 run, however, both simulations show the correct trend in spite of a few exceptions. The following are the general conclusions derived from the MADCAP simulation for San Diego Air Basin.

- MADCAP correctly predicts ozone peak and decay; ozone peaks right after mid-noon and decays toward the latter part of the day.
- MADCAP correctly predicts the trend of NO<sub>x</sub> and hydrocarbon species; NO concentration is high in the morning, decreases in the daytime and shows a minimum at midday.
- Whenever predicted NO concentration exceeds measured NO concentration, it is observed that predicted O<sub>3</sub> concentration is less than measured ozone concentration. NO trends (predicted vs measured) similarly correspond to ozone trends.
- Without terrain, the MADCAP wind field compares well with the WEST prediction. Wind speed agrees to about 5% whereas the wind pattern agrees within 20%.
- Simulation results are strongly influenced by the boundary conditions. The trends predicted by the small regional simulation are similar to the trends predicted by the large regional simulation. The small regional simulation has larger boundary effects than the large regional simulation.
- MADCAP is sensitive to the vertical resolution often showing ground level concentrations to be inversely proportional to the spacing of the first level. Fifty meters appears to be a reasonable spacing for MADCAP applications to San Diego.
- MADCAP is not very sensitive to the number of steps per hour. Ten steps per hour appears more than adequate to treat the San Diego 2 Km source grid.

- MADCAP is sensitive to mixing height changes only if the inversion base is within 25 meters of the ground.
- Doubling the cell resolution from  $4 \text{ Km}^2$  to  $16 \text{ Km}^2$  resulting in at most a 20% decrease in calculated values.
- The MADCAP baseline formulation is not sensitive to diffusion changes.
- Ground level concentration is found to be very sensitive to the manner in which diffusion is calculated. One half hour post-diffusion after source injection results in considerably lowered concentrations and consequent under predictions relative to measured values. For San Diego County, the sensitivity runs indicate that 15 minutes of diffusion after source injection would probably yield best agreement with measured data.
- Stream tube photochemical sensitivity analyses show strong sensitivity because a single Lagrangian fluid element is used.
- Hydrocarbons are effective precursors of ozone, but not linearly. Hence, linear rollback is not effective as a predictive scheme (as found by many others).
- Increased NO levels tend to reduce ozone.
- Decreased NO levels allow more rapid ozone formation as well as higher ozone levels.
- Olefins (HC1) and particularly Oxygenates (HC3) are most effective reactive precursors of ozone.
- Alkanes (HC2) are ineffective as  $O_3$  precursors.
- $NO_2$  is effective as an  $O_3$  precursor because  $NO_2$  can be photo-dissociated to provide atomic oxygen to generate ozone.
- Insolation is the most sensitive parameter.
- Interaction between stream tubes are very important as shown the differences observed between the single stream tube sensitivity study and full regional sensitivity study.
- Small sensitivities to the emission of  $NO_x$ , total hydrocarbons and splitting of the hydrocarbon species are observed for the San Diego Air Basin in the regional simulation. This is because total source volume emitted over a day is small relative to the total holding volume of the San Diego Air Basin. This is indicative of low pollution levels in San Diego.

- In contrast, boundary and initial conditions show great sensitivity because they are representative of and commensurate with total atmospheric pollution.

In general, we feel that MADCAP is a physically well-organized program developed to study the photochemical effects in atmosphere. However, improvements are possible. The following are a few recommendations which should be considered for the current version of MADCAP model.

- A sensitivity study on the different diffusion schemes should be performed to evaluate the model predictions.
- Terrain data should be included in the determination of the wind field.
- The wind field generated in a Lagrangian cell (where the vertical layer is conformed to the ground level) should be compared with the wind field generated in an Eulerian cell with terrain effect.
- Because of the relatively lengthy running time (on the order of hours), a restart capability should be incorporated into the model.
- Other improvements such as time centering insolation routine, etc., can be made to expedite the running time and increase the model accuracy.
- Since the model is sensitive to initial and boundary conditions, special care should be given to deriving the initial and boundary conditions.
- Due to the low level of San Diego emission, no significant sensitivity to  $\text{NO}_x$  and hydrocarbon in the regional simulation is observed. It is felt that MADCAP should be validated and studied in some other heavily polluted area such as the Los Angeles Air Basin before its general usage as a predicting tool.

## REFERENCES

1. Fabrick, A.J., R.C. Sklarew, and J.C. Wilson, "Point Source Model Evaluation and Development Study", Contract A5-058-87 for the California Air Resources Board and California Energy Resources Conservation and Development Commission, Science Applications, Inc., Westlake Village, California, 91361, March 1977.
2. Martinez, J.R., R.A. Nordsieck, and M.A. Hirschberg, "User's Guide to Diffusion/Kinetics (DIFKIN) Code Final Report", EPA-R4-73-012, General Research Corporation, Santa Barbara, California, 93105; December 1973.
3. Gelinas, R.J., and P.D. Skewes-Cox, "Troposphere Photochemical Mechanisms", Journal of Physical Chemistry, Vol 81, No. 25, pp 2468 - 2479, 1977.
4. Troijonis, J.C., and K.W. Arledge, "Utility of Reactivity Criteria in Organic Emission Control Strategies for Los Angeles", Final Report of EPA Contract No. 68-02-1735, December 1975.
5. Turner, D.B., "Workbook of Atmospheric Dispersion Estimate", Environmental Protection Agency (Revised), 1970.
6. Myrup, L.O., and A.J. Ranzieri, "A Consistent Scheme for Estimating Diffusivities to be Used in Air Quality Models: Interim Report", California Department of Transportation, June 1976.
7. Eschenroeder, A.Q., "Evaluation of a Diffusion Model for Photochemical Smog Simulation", Environmental Protection Agency Publication EPA-R4-73-012, 1972.
8. Hecht, T.A., J.H. Seinfeld, and M.C. Dodge, "Further Development of Generalized Kinetic Mechanism for Photochemical Smog", Environmental Science and Technology, Vol. 8, No. 4, pp 327 - 339, 1974.
9. Sklarew, R.C., A.J. Fabrick, and J.C. Wilson, "DEPICT, Formulation and User's Manual", Science Applications, Inc., La Jolla, California, 92037, January 1976.

10. Wilson, J., Private Communication:

The actual hydrocarbon sources were taken from the report entitled :Source Reconciliation of Atmospheric Hydrocarbon in South Coast Air Basin, 1974" by Mayrsohn, Cabtree, Kuramoto, Sothern and Mano, CARB, Division of Technical Service, DTS-76-2 (July, 1975). Three different regression analyses were performed to determine the total splitting factors of hydrocarbon: pre-dawn analysis, morning analysis and midday analysis. It is found that the total splitting factors vary insignificantly with time.

**Appendix A**  
**MADCAP USER's MANUAL**

## Appendix A

### MADCAP USER'S MANUAL

#### A.1 PRE-PROCESSORS USER'S MANUAL

The pre-processors PREBIC, PREMET and PRESOR read in various geographic, air quality, meteorological and emissions data from a common file assigned the logical unit number 5. In addition, PRESOR requires the extrapolated wind field and stability data generated by PREMET on a file with logical unit number 14. In section A.1.1 below, the common data file of unit 5 is described. In section A.1.2, A.1.3 and A.1.4 is a further look at I/O and the various controlling parameters of PREBIC, PREMET and PRESOR, respectively. Finally, section A.1.5 contains more general warnings and recommendations regarding the pre-processors collectively.

##### A.1.1 Input Deck

All three pre-processors read in data from a common file assigned logical unit number five. This file contains the raw data as collected on site, and each data record is 80 bytes in length and identified by a two digit number in the first two bytes. The structure of the input deck is described in the following list:

<u>CARD TYPE</u>	<u>DESCRIPTION</u>
n/a	Namelist SETUP
n/a	Namelist GRID0
n/a	Namelist TERRO
72	Average aerodynamic roughness height by grid cell.
73	Average land use category by grid cell.
31	Ground station air quality.

<u>CARD TYPE</u>	<u>DESCRIPTION</u>
32	Ground station air quality.
41	Temperature soundings.
42	Winds aloft (pibal).
81	Motor vehicle emissions.
82	Shipping emissions.
83	Railway emissions.
84	Area source emissions.
85	Ground level point source emissions.
91	Elevated point source emissions.
93	Aircraft emissions.

For an N hour run, there should be N+1 files merged onto logical unit five. File 1 contains the three namelists and data of type 72 and 73 (in this order). Note that the data of file 1 is not a function of time of day but depends only on grid parameters. The remaining N files contain chronologically ordered hourly data (hour 00 representing the span midnight to 1 am). An hourly file consists of data of types 31, 32, 41, 42, 81 - 85, 91, 93 in numerical order. The first card of each card type is blank except for its type identifier.

Embedded decimals are found on card types 84 and 85 only. These fields are shown as N 3.3, this indicates that the decimal is in front of the first digit (i.e. a 122 should be read as .122). A variable listed as an N 2 means that a 2 digit integer will occupy that field. An N 2.0 means real number will occupy the 2 digit field. If a decimal does not appear, it is implied at the end of the field.

The "run number" shown on each record documents the assumptions used to generate the output from the Modeling Studies Data Staging (MSDS) system.

When data are missing for any particular fields, a value of -1 will be entered into the data field.

The following pages give complete data descriptions by namelist and card type. Note that all spatial coordinates of the input data are relative to the full 50 x 60 grid.

INPUT DESCRIPTION  
PRE-PROCESSORS  
NAMELIST SETUP

<u>VARIABLE</u>	<u>DESCRIPTION</u>
MONTH	Month of start of simulation
DAY	Day of start of simulation
YEAR	Year of start of simulation
BEGIN	First hour of simulation
NOHRS	# of hours of simulation

INPUT DESCRIPTION  
PRE-PROCESSORS  
NAMELIST GRID0

<u>VARIABLE</u>	<u>DESCRIPTION</u>
MX	Desired # cells in X direction + 1 for simulation
MY	Desired # cells in Y direction + 1 for simulation
MZ	Desired # cells in Z direction + 1 for simulation
MIX	Desired # cells in X direction between the origin of the 50 x 60 grid and the origin of the MX x MY grid
MIY	Desired # cells in Y direction between the origin of the 50 x 60 grid and the origin of the MX x MY grid

Note: A simulation run of MADCAP is always defined relative to a fixed 50 x 60 grid containing all geographical areas of interest. Thus, any given simulation run is contained in a pre-determined 50 x 60 grid.

INPUT DESCRIPTION  
PRE-PROCESSORS  
NAMELIST TERRO

<u>VARIABLE</u>	<u>DESCRIPTION</u>
IBC1(60)	Relative to the 50 x 60 grid, IBC1(I) holds the x-ordinate of the mountain boundary at y ordinate I
IC1(60)	Relative to the 50 x 60 grid, IC1(I) holds the x-ordinate of the coastal boundary at y-ordinate I.

\*Note: for S.D.

IBC1 = 10 \* 37, 2 \* 40, 2 \* 39, 37, 2 \* 36, 2 \* 39, 40, 5 \* 41,  
2 \* 35, 8 \* 37, 34, 33, 2 \* 31, 30, 4 \* 27, 3 \* 33, 32,  
31, 2 \* 30, 29, 28, 27, 6 \* 25

IC1 = 10 \* 25, 2 \* 24, 23, 7 \* 19, 2 \* 18, 7 \* 19, 3 \* 18,  
3 \* 17, 3 \* 16, 15, 2 \* 14, 13, 2 \* 12, 11, 2 \* 10, 9,  
8, 7, 6, 5, 4, 3, 2, 5 \* 1

## INPUT DESCRIPTION

## PRE-PROCESSORS

AVERAGE AERODYNAMIC ROUGHNESS HEIGHT

72

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code Identifying type of input '72'
3	N 3	East-West 'I' Cell Coordinate*
6	N 3	Starting North-South 'J' Cell Coordinate
9	N 3	Ending North-South 'J' Cell Coordinate
12	(10) N 4.0	Aerodynamic Roughness Height (Meters)
52	25	Unused Filler
77	N 4	Run Number

\* All I and J coordinates for all data types on unit five  
are relative to the full 50 x 60 grid.

## INPUT DESCRIPTION

## PRE-PROCESSORS

## AVERAGE LAND USE CATEGORY

73

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code Identifying type of input '73'
3	N 3	East-West 'I' Cell Coordinate
6	N 3	Starting North-South 'J' Cell Coordinate
9	N 3	Ending North-South 'J' Cell Coordinate
12	(10) N 4.0	Land Use Category (see Table 1)
52	25	Unused Filler
77	N 4	Run Number

<u>Land Use Category</u>	<u>Form G01 Code</u>
Low density residential	1
High density residential	2
Office buildings	3
Central business district	4
Shopping center	5
Industrial	6
Orchards	7
Other agricultural	8
Forests	9
Shrub land	10
Open water	11
Parks densely foliated	12
Other parks	13

## INPUT DESCRIPTION

## PRE-PROCESSORS

## GROUND STATION AIR QUALITY

31

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '31'
3	N 3	East-West 'I' Cell Coordinate
6	N 3	North-South 'J' Cell Coordinate
9	N 2	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period applies to 00-23 Pacific Standard time
17	N 4.0	Measured pollutant concentration of CO (PPM)
21	N 4.0	Measured pollutant concentration of NO <sub>x</sub> (PPM)
25	N 4.0	Measured pollutant concentration of NO <sub>2</sub> (PPM)
29	N 4.0	Measured pollutant concentration of NO (PPM)
33	N 4.0	Measured pollutant concentration of SO <sub>2</sub> (PPM)
37	N 4.0	Measured pollutant concentration of Partic- ulates (PPM)
41	N 4.0	Measured pollutant concentration of Methane (PPM)
45	N 4.0	Measured pollutant concentration of Non Methane (PPM)
49	N 4.0	Measured pollutant concentration of Total HC (PPM)
53	N 4.0	Measured pollutant concentration of Ozone(PPM)
57	20	Unused Filler
77	N 4	Run Number

## INPUT DESCRIPTION

## PRE-PROCESSORS

## GROUND STATION METEOROLOGY

32

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '32'
3	N 3	East-West 'I' Cell Coordinate
6	N 3	North-South 'J' Cell Coordinate
9	N 2	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period data applies to 00-23 Pacific Standard time
17	N 4.0	Ambient Temperature (Degrees C)
21	N 4.0	Relative Humidity (Percent)
25	N 4.0	Wind Direction Azimuth (Degrees from North)
29	N 4.0	Wind speed (Meters/sec)
33	N 4.0	Cloud Cover NWS definition (Percent) 0 percent indicates no cloud cover
37	N 4.0	Height of cloud cover above ground (meters) 9999 indicates no cloud cover
41	N 6.0	Ambient Pressure (millibars)
47	N 6.0	Solar Insolation (watts/M <sup>2</sup> )
53	24	Unused Filler
77	N 4	Run Number

INPUT DESCRIPTION  
PRE-PROCESSORS  
TEMPERATURE (INVERSION) SOUNDINGS

41

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '41'
3	N 3	East-West 'I' Cell coordinate
6	N 3	North-South 'J' Cell coordinate
9	N 2	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period data applies to 00-23 Pacific Standard time
17	N 4	Ground elevation of measurement location above MSL (meters)
21	N 4	Inversion base above Mean Sea Level (meters)
25	N 2	Ambient Temperature at Base (degrees C)
27	N 4	Relative Humidity at Base (percent)
31	12	Unused Filler
43	N 4	Inversion top above Mean Sea Level (meters)
47	N 2	Ambient temperature at Top (degrees C)
49	N 4	Relative Humidity at Top (percent)
53	24	Unused Filler
77	N 4	Run Number

## INPUT DESCRIPTION

## PRE-PROCESSORS

WINDS ALOFT  
(PIBAL)

42

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '42'
3	N 3	East-West 'I' Cell coordinate
6	N 3	North-South 'J' Cell coordinate
9	N 2	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period data applies to 00-23 Pacific Standard time
17	N 2	Number of records remaining for a sounding including this record
19	2	Unused Filler
21	N 4	Elevation of Pibal above Mean Sea Level (meters)
25	N 4	Elevation of station above MSL (meters)
29	2	Unused Filler
31	N 4	Wind Direction Azimuth (degrees from north)
35	N 4	Wind speed (meter/sec)
39	38	Unused Filler
77	N 4	Run Number

INPUT DESCRIPTION  
PRE-PROCESSORS  
MOTOR VEHICLE EMISSIONS

81

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '81'
3	N 3	East-West 'I' Cell coordinate
6	N 3	North-South 'J' Cell coordinate
9	N 2	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period data applies to 00-23 Pacific Standard time
17	N 6.0	Flux of CO (kg/hr)
23	N 6.0	Flux of NOx (Kg/hr)
29	N 6.0	Total Flux of THC (Kg/hr)
35	N 3.0	Gasoline powered vehicle exhaust*
38	N 3.0	Gasoline powered vehicle evaporation*
41	N 3.0	Diesel powered vehicle exhaust*
44	33	Unused Filler
77	N 4	Run Number

\* Source category fractions of the total THC for source listed.

## INPUT DESCRIPTION

## PRE-PROCESSORS

## SHIPPING EMISSIONS

82

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '82'
3	N 3	East-West 'I' Cell coordinate
6	N 3	North-South 'J' Cell coordinate
9	N 2	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period data applies to 00-23 Pacific Standard time
17	N 6.0	Flux of CO (Kg/hr)
23	N 6.0	Flux of NOx (Kg/hr)
29	N 6.0	Flux of THC (Kg/hr)
35	42	Unused Filler
77	N 4	Run Number

## INPUT DESCRIPTION

## PRE-PROCESSORS

## RAILWAY EMISSIONS

83

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '83'
3	N 3	East-West 'I' Cell coordinate
6	N 3	North-South 'J' Cell coordinate
9	N 2	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period data applies to 00-23 Pacific Standard time
17	N 6.0	Flux of CO (Kg/hr)
23	N 6.0	Flux of NOx (Kg/hr)
29	N 6.0	Flux of THC (Kg/hr)
35	42	Unused Filler
77	N 4	Run Number

INPUT DESCRIPTION  
PRE-PROCESSORS

84  
or  
85

AREA SOURCE EMISSIONS & GROUND LEVEL POINT SOURCE EMISSIONS

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '84 or 85'
3	N 3	East-West 'I' Cell coordinate
6	N 3	North-South 'J' Cell coordinate
9	N 2	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period data applies to 00-23 Pacific Standard time
17	N 6.0	Flux of CO (Kg/hr)
23	N 6.0	Flux of NOx (Kg/hr)
29	N 6.0	Total Flux of THC (Kg/hr)
35	N 3.3	Dry cleaning: petrol based*
38	N 3.3	Dry cleaning: synthetic*
41	N 3.3	Gasoline marketing: underground storage*
44	N 3.3	Gasoline marketing: auto tank filling*
47	N 3.3	Waste burning and fires*
50	N 3.3	Surface coating: air dried archit*
53	N 3.3	Pesticides*
56	N 3.3	Fuel combustion*
59	N 3.3	Surface coating: heat treated*
62	N 3.3	Surface coating: air-dried industrial*
65	N 3.3	Degreasing: halogenated*
68	N 3.3	Degreasing: non-halogenated*
71	N 3.3	Industrial process sources*
74	3	Unused Filler
77	N 4	Run Number

## INPUT DESCRIPTION

## PRE-PROCESSORS

## ELEVATED POINT SOURCE EMISSIONS

91

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '91'
3	N 3	East-West 'I' Cell coordinate
6	N 3	North-South 'J' Cell coordinate
9	N 3	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period data applies to 00-23 Pacific Standard time
17	N 6.0	Flux of CO (Kg/hr)
23	N 6.0	Flux of NOx (Kg/hr)
29	N 6.0	Flux of THC (Kg/hr)
35	N 5.0	Height of stack above ground (meters)
40	N 5.0	Diameter at top of stack (meters)
45	N 5.0	Exit gas velocity from stack (meter/sec)
50	N 5.0	Exit gas temperature (degrees C)
55	N 5.0	Volume flow rate of gas (meter <sup>3</sup> /sec)
60	17	Unused Filler
77	N 4	Run Number

INPUT DESCRIPTION  
PRE-PROCESSORS  
AIRCRAFT EMISSIONS

93

<u>RECORD POSITION</u>	<u>FORM &amp; LENGTH</u>	<u>DESCRIPTION</u>
1	N 2	Code identifying type of input '93'
3	N 3	East-West 'I' Cell coordinate
6	N 3	North-South 'J' Cell coordinate
9	N 2	Year (last 2 digits) of study period
11	N 2	Month of study period
13	N 2	Date of study period
15	N 2	Hour of study period
17	N 6.0	Flux of CO (Kg/hr)
23	N 6.0	Flux of NOx (Kg/hr)
29	N 6.0	Total Flux of THC (Kg/hr)
35	N 3.0	Fraction of THC from piston aircraft
38	N 3.0	Fraction of THC from jet aircraft
41	20	Unused Filler
61	N 4.0	Mean aircraft altitude above ground (meters)
65	12	Unused Filler
77	N 4	Run Number

### A.1.2 PREBIC

PREBIC reads data from logical unit five as described in Section A.1.1 and writes both to unit 10 as input to MADCAP and to unit 6 as a summary of the input boundary and initial conditions. PREBIC requires only the three namelists and the data of type 31, though a deck prepared in accordance with A.1.1 is recommended. Onto unit six, PREBIC echoes the input data type 31, prints the mountain boundaries, plots and prints the level 1 concentrations of CO, O<sub>3</sub>, NO<sub>2</sub>, NO, HC1 & HC2 for the first hour of simulation (the initial conditions), and hourly concentrations along the boundary of level 1 for the same species (the boundary conditions). Onto unit 10, PREBIC writes only the initial conditions and hourly boundary conditions.

The only data to be set internally in PREBIC are the variables MS and NS. MS is the number of major reactive species (for the coded kinetics, the major reactive species are CO, O<sub>3</sub>, NO<sub>2</sub>, NO, HC1, HC2 and HC3, hence MS = 7). NS is the number of pollutant species recorded on the data cards of type 31 (at present, the species input for San Diego are CO, NO<sub>x</sub>, NO<sub>2</sub>, NO, SO<sub>2</sub>, particulates, CH<sub>4</sub>, non-methane hydro-carbons (NMHC), total hydro-carbons (THC), and O<sub>3</sub>, hence NS = 10).

The dimensions that should be modified from run to run of PREBIC are described in the following list (MX, MY, MZ as in A.1.1, MS and NS as above, MAXSITES = total number of data sites for card type 31).

IC (MY), IX (MAXSITES), JY (MAXSITES), C1 (NS, MAXSITES), CMAX (MS), CMIN (MS), CON (MS), ANORM (MS), NAME (MS), IBC (MY), C (NS, MAXSITES), CE (MX, MY, MS), BCX (MS, MY, 2), BCY (MS, MX, 2) II (MAXSITES), JJ (MAXSITES).

### A.1.3 PREMET

PREMET reads from logical unit 5 as described in Section A.1.1 and writes onto unit 11 as input to MADCAP, onto unit 14 as input to PRESOR, and onto unit 6 as summary of the meteorological conditions. PREMET requires only the three namelists and the data of types 72, 73, 32, 41 and 42, though a deck prepared in accordance with A.1.1 is recommended. Onto unit 6, PREMET plots and prints land use and aerodynamic roughness, then for each hour echoes the ground level input data, plots and prints all of cloud cover, ceiling height, temperature, relative humidity, Turner stability class,  $H_2O$  concentration and solar intensity on the ground level. Then (still onto unit 6 for each hour) PREMET echoes the inversion data, plots and prints the inversion top and base, plots and prints the diffusivities on all vertical levels, echoes ground station wind data, echoes pibal data, and finally prints the wind field for each vertical level. Onto unit 14, PREMET writes the ground level wind speed and Turner stability class profiles. Onto unit 11, PREMET writes all of solar intensity, water concentration, the multi-level diffusivities and the multi-level wind field.

The only data to be set internally in PREMET consists of the latitude (ALAT) and longitude (ALONG) represented in the solar insolation calculation, a flag for the time zone (BRING) and an array that holds the heights (in meters) of the vertical levels (ZC). BRING = 75 for EST, 90 for CST, 105 for MST, and 120 for PST. The array ZC holds the Z-coordinates of the cell-centers. Thus, ZC(I) is the Z-coordinate of the cell-center of the I<sup>th</sup> cell up from the ground level.

The dimensions that should be modified from run to run of PREMET are described in the following list (MX, MY, MZ as in A.1.1,

NMET = 1 + MAX # of met cards/hr (type 32)

NINV = 1 + MAX # of inversion cards/hr (type 41)

NPIB = 1 + MAX # of pibal sites/hour

NPEL = MAX # of pibal elevations

V(MX, MY, MZ), U (MX, MY, MZ), II (NMET), JJ (NMET), TEMP (NMET), RELHUM (NMET), WD (NMET), CC (NMET), CH (NMET), UV (NMET), IX (NMET), JY (NMET), IBC(MY), ZC (MZ), I41 (NINV), J41 (NINV), ZBOT (NINV), TBOT (NINV), ZTOP (NINV), TTOP (NINV), TGRAD (NINV), P(NINV), DMAX (MZ), DMIN (MZ), WDV (NPEL), WSV (NPEL), IIX (NPIB), JJY (NPIB), CLOUD (MX, MY), CEIL (MX, MY), T (MX, MY), RH (MX, MY), WS (NMET, MZ), ZIBSE (MX, MY), ZITOP (MX, MY), KSTAB (MX, MY), SOL (MX, MY), H2O (MX, MY), TT (MX, MY), DTDZ (MX, MY), PWS (NPIB, MZ), PWD (NPIB, MZ), D (MX, MY, MZ), SPEED (MX, MY).

It should be mentioned that only the first two UV data per hour will be processed by PREMET.

#### A.1.4 PRESOR

PRESOR reads data from both logical unit 5 as described in Section A.1.1 and unit 14 as generated by PREMET. PRESOR writes both to unit 12 as input to MADCAP and to unit 6 as a summary of the pollutant sources. PRESOR requires only the three namelists and the data of types 81-85, 91 and 93. though a deck prepared in accordance with A.1.1 is recommended. Onto unit 6 for each hour for each source type, PRESOR plots and prints CO, NO<sub>2</sub>, NO, HC1, HC2 and HC3 fluxes, and then plots and prints total fluxes over all sources for each source type for each hour. Onto unit 12 are written hourly fluxes for each species over all source types.

The only data to be set internally in PRESOR consists of a height (in meters) array (HT), and a flag defining which plume rise option is to be used (IFLGG) and the DX, DY of the grid. HT(I) contains the height above ground of the upper boundary of level I, and IFLGG = 1 specifies Briggs 67, = 2 species Briggs 74, and = 3 specifies TVA.

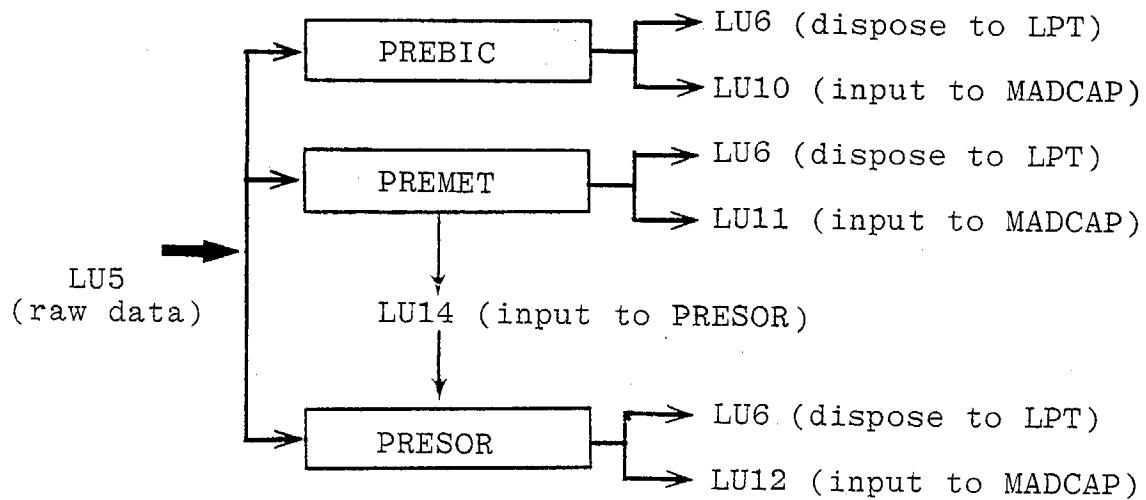
The dimensions that should be modified from run to run of PRESOR are described in the following list (MX, MY, MZ as in A.1.1).

STORE (MX, MY, 6), SE (6, MX, MY, MZ), S (MX, MY, 6), DZ (MZ), HT (MZ), IBC (MY), WIND (MX, MY), KSTAB (MX, MY).

#### A.1.5 Warnings and Recommendations Regarding the Pre-Processors

1. The dimension changes discussed in Sections A.1.2, A.1.3 and A.1.4 should be made even if the set dimensions seem sufficiently large for the actual MX, MY, MZ, etc. This is because the data transfer between the pre-processors and MADCAP is unformatted, and in fact the data transfers to the printing and plotting routines require memory to be stacked properly.
2. The arrays IBC1 and IC1 in the namelist TERRO should be set even if there is no terrain boundary. That is to say, that if there is no ocean boundary, set IC1 identically equal to 1, and if there is no mountain boundary, set IBC1 identically equal to 50. This is the correct and appropriate way to input the boundaries under these conditions. It is assumed that the coast boundary, if present, lies to the west (smaller I value) of the mountain boundary, if present, and neither boundary may run along the northern or southern edge of the grid.

- 3) The I/O of the pre-processors is somewhat confusing. Thus, a flow-chart style diagram of the I/O is included below:



- 4) The arrays ZC and HT in PREMET and PRESOR, respectively, must be consistent. ZC (I) is the Z coordinate of the center of cell I, while HT is the Z coordinate of the top of cell I.
- 5) As indicated in section A.1.1, any given run of the preprocessors is defined on a subset of a 50 x 60 grid. The co-ordinates of the data as input to the pre-processor are relative to the 50 x 60 grid, as are the arrays IBC1 and IC1. The parameters defining the subgrid of interest are in the namelist GRID0.

## A.2 MADCAP USER'S MANUAL

### A.2.1 Madcap Input/Output

The program MADCAP reads in the output of PREBIC, PREMET and PRESOR from logical units 10, 11, and 12, respectively. In addition, a file assigned to logical unit 5 should contain the namelists GRID0 and TERRO in that order (as described in A.1.1), followed by the record:

NPLACE, NOHRS, NHR, NDATE, IPRTSP  
formatted as (4A4, 4012). NPLACE(4) is the region for which the simulation is to be run, NOHRS is the duration (in hours) of the simulation, NDATE (1) is the month of the simulation (e.g. JAN is represented by 01, etc.), NDATE (2) is the day of the simulation, and NDATE (3) is the year of the simulation (e.g. 75, 69 etc.). The array IPRTSP defines which species concentrations will be plotted and printed on paper. IPRTSP (I) = 01, signifies that the species I concentrations should be output; IPRTSP (I) = 00 signifies otherwise. The species are, as usual, ordered as CO, O3, NO2, NO, HC1, HC2, HC3. Thus, to have hourly plots and prints of NO2 concentration profiles, set IPRTSP (3) = 01; if NO2 concentration output is not required, set IPRTSP (3) = 00.

MADCAP writes a file on logical unit 6 consisting of hourly plots and prints of concentration profiles of the species for which IPRTSP = 01. These plots and prints are made for each vertical level. The output of each hour is followed by a listing of the maximum hourly concentration for each major reactive species and the hourly output is followed by hourly concentrations for all levels for the locations given in the array IX, JY (See Section A.2.2). Note that these last outputs are independent of IPRTSP.

If an unformatted disk file of hourly concentrations is desired, the parameter MYDSK should be set to 1 in the data statement of MADCAP. Otherwise set MYDSK = 0. The unformatted file is written onto logical unit 13. The output onto 13 for each hour consists of 1 formatted record describing location, 1 formatted record containing the hour and date (these two records are best read as titles in Hollerith mode) and then the unformatted concentration array.

The order of indices of the output concentration array is described by:

CONCENTRATION (I, J, K, L)

where

I = x - coordinate

J = y - coordinate

K = z - coordinate

L = species # (ordered CO, O3, NO2, NO, HC1, HC2, HC3, N2O5, HNO3, HNO2, HO2, RN02, H2O2).

Since the output of a typical run of MADCAP is quite voluminous, an option has been included which allows a quick perusal of the results. If the parameter NEDIT is set to 1 in the data statement of MADCAP a file is written onto logical unit 8 which contains only the NO and O3 concentration profiles. These data should give some sense of the simulation. If NEDIT = 0, nothing is written onto logical unit 8.

#### A.2.2 MADCAP Data and Dimensions

The data to be set internally in MADCAP consists of the switches MYDSK and NEDIT, the grid parameters DX1, DY1, DZ1, the parameter NOPL and the arrays PLACE, IX, and JY. MYDSK = 1 generates an unformatted file of hourly concentratins on logical unit 13 (See Section A.2.1); otherwise set MYDSK = 0. Similary, NEDIT = 1

generates a formatted summary file on logical unit 8 for quick perusal (See Section A.2.1); otherwise set NEDIT = 0. DX1, DY1 are the DX, DY (in meters) of the grid, respectively. DZ1 is an array that holds the vertical zoning. DZ1(1) is the height (in meters) of the ground level cell. Similarly, DZ1(I) holds the height (in meters) of the cell of level I. Thus, if the ground cell has center at 5 m, the next cell up has center at 15 m, and the 3rd cell up from the ground has center at 75 m, then DZ1(1) = 10, DZ1(2) = 10, and DZ1(3) = 110. The arrays PLACE, IX and IY hold the locations for which separate output is desired, where PLACE(I) is an 8 byte hollerith name for location I with x - coordinate IX(I) and y coordinate JY(I). The scalar NOPL should be set to the number of sites of special interest. The additional output consists of numerical values of major reactive species concentrations for each hour for each level and follows the last hourly output.

The dimensions that require modification are described in the following list. The changes in the blank common should be made in every appearance of the common (MS = # of primary reactive species, NS = total # of reactive species MX, MY and MZ are as in A.1.1, NOHRS = # hours of simulation, NOPL = # locations of special interest and INT is the number of chemistry integration steps per hour plus one; note that for the coded kinetics MS = 7, NS = 13)

COMMON NVERS (6), NDCOD1, NDCOD2, NDCOD3, NPLACE (4),  
NDATE (3), NHR, NOHRS, MX, MY, MZ, MS, NS, DX, DY, NSTEP, AVESOL,  
UF, UFX, DI, LI, LIS, UFY, INT1, INT, MXY, MXYZ, MXYZS, MXS, MYS,  
MXYS, MXZ, MYZ, MXZS, MYZS, DZ (MZ), CE (MX, MY, MZ, NS), SE (MS,  
MX, MY, MZ), IBC (MY), BCX (NS, MY, MZ, 2), BCY (NS, MX, MZ, 2),  
SOL (MX, MY), CEG (MX, MY, MS), BCXG (MS, MY, 2), BCYG (MS, MX, 2),  
U (MX, MY, MZ, 2), D (MX, MY, MZ), H2O (MX, MY), NSPEC (2, MS),  
IPRTSP ( MS), UFZ2 (MZ), XMAX (NS, 2), SL ( MS, INT), CC ( NS),  
CL (MX, MY, MZ, NS), XMIN (NS), ICOAST (MY).

NSPEC1 (2, MS), DZ1 (MZ), IX (NOPL), JY (NOPL), PLACE  
(NOPL), STORE (NOMRS, NOPL, MZ, MS).

### A.3 CODE LISTING

#### A.3.1 PREBIC

```
00001 C THIS DRIVES PREBIC WITH THE ASSUMPTION THAT THE
00002 C FIRST FILE ON FT05F001 IS THE NAMELIST SSETUP,
00003 C THAT THE SECOND FILE IS THE NAMELIST SGRID0,
00004 C THAT THE THIRD FILE IS THE NAMELIST STERR0,
00005 C THAT THE FOURTH FILE IS THE INVARIANT DATA,
00006 C THAT THE FIFTH FILE IS THE HOURLY DATA FOR THE
00007 C FIRST HOUR TO BE PROCESSED FOLLOWED BY CONSECUTIVE
00008 C HOURLY FILES
00009 C
00010 C
00011 C      DIMENSION NDATE(3), IBC1(60), IC1(60)
00012 C      INTEGER DAY, YEAR, BEGIN
00013 C      COMMON/TERRA/ IBC1, IC1
00014 C      COMMON/GRIDA/MX, MY, MZ, MIX, MIY
00015 C      NAMELIST/GRID0/MX, MY, MZ, MIX, MIY
00016 C      NAMELIST/TERR0/IBC1, IC1
00017 C      NAMELIST /SETUP/ MONTH, DAY, YEAR, BEGIN, NOHRS
00018 C
00019 C
00020 C      READ(5, SETUP)
00021 C      WRITE(6, SETUP)
00022 C      READ(5, GRID0)
00023 C      WRITE(6, GRID0)
00024 C      READ(5, TERR0)
00025 C      WRITE(6, TERR0)
00026 C      NDATE(1) = MONTH
00027 C      NDATE(2) = DAY
00028 C      NDATE(3) = YEAR
00029 C      IHR = BEGIN
00030 C      DO 10 N = 1, NOHRS
00031 1     READ(5, 101) ITYPE
00032 C      IF( ITYPE.NE.31) GO TO 1
00033 C      CALL PREBIC(NDATE, IHR)
00034 C      IHR = IHR + 1
00035 C      IF( IHR.LT.24) GO TO 10
00036 C      IHR = 0
00037 C      NDATE(2) = NDATE(2) + 1
00038 10    CONTINUE
00039 C      STOP
00040 101   FORMAT( I2)
00041 C      END
```

```

00001      SUBROUTINE PREBIC(NDATE,NHR)
00002      C      CREATES BOUNDARY CELL CONCENTRATIONS AND CE INITIAL HOUR CONCENTRATI
00003      C      ONS
00004      C      ALL OUTPUT (TAPES10,11,12) IS RELATIVE TO SMALL GRID (MX,MY,MZ)
00005      C      COMMON MIX,MIY,MX,MY,MZ, IBC(MY),C(NS,MAXSITES),NHR,NDATE(3),
00006      C      1CE(MX,MY,MS),           C1(MS,MAXSITES),BCX(MS,MY,2),BCY(MS,MX,2),
00007      C      2IX(MAXSITES),JY(MAXSITES),II(MAXSITES),JJ(MAXSITES),RSQ(MX,MY,MAXS
00008      C      3CMAX(MS),CMIN(MS),      RSQ(MX,MY,MAXSITES)
00009      C      4CD(MXY,MS),
00010      C      5
00011      C      6MXY
00012      C      DIMENSION IBC1(60),IX(40),JY(40),IC1(60), C1(10,40),CMAX( 7)
00013      1,CMIN( 7),CON(7),ANORM(7)
00014      C      DIMENSION          IBC(51),C(10, 40),      NDATE(3),
00015      1CE(37,51, 7),           BCX( 7,51,2),BCY( 7,37,2),
00016      2,          II( 40),JJ( 40),      MDATE(3),
00017      3NDCOD(2,3)
00018      INTEGER NAME(7), IC(51)
00019      COMMON/TERRA/IBC1,IC1
00020      COMMON/GRIDA/MX,MY,MZ,MIX,MIY
00021      EQUIVALENCE (C,IC1)
00022      DATA NAME    /' CO ',' 03',' NO ',' NO ',' HC1',' HC2',' HC3'/
00023      M=0
00024      NS=10
00025      MS=7
00026      NDCOD(1,1)=1
00027      NDCOD(2,1)=1
00028      MXY=MX*MY
00029      DO 27 NN=1,MY
00030      IC(NN)=MAX0( IC1(NN+MIY)-MIX, 1)
00031      27 IBC(NN)= MIN0( IBC1(NN+ MIY)- MIX, MX)
00032      C      READ AQ DATA (CARD TYPE 31)
00033      C      READ(7)
00034      1 M=M+1
00035      READ(5,101) ITYPE,II(MD,JJ(MD,MDATE(3),MDATE(1),MDATE(2),
00036      1MHR,(C(L,MD,L=1,10)
00037      101 FORMAT( 12,2I3,4I2, 10F4.0 )
00038      C      WRITE(6,11112) ITYPE,M,II(MD,JJ(MD,MDATE(3),MDATE(1),MDATE(2),MHR,
00039      C      1C(L,MD,L=1,10)
00040      C11112 FORMAT(///(1X,G10.2)/)
00041      C      CHECK CARD TYPE
00042      IF( ITYPE.NE.31.AND. ITYPE.NE.32) GO TO 900
00043      IF( ITYPE.EQ.32) GO TO 3
00044      C      CHECK DATE
00045      IF( MHR.EQ.NHR.AND. MDATE(1).EQ.NDATE(1).AND. MDATE(2).EQ.NDATE(2).A
00046      1ND.MDATE(3).EQ.NDATE(3)) GO TO 2
00047      C      MESSAGE ABOUT WRONG TIME
00048      IAM=2
00049      WRITE(6,102) MHR,MDATE,NHR,NDATE,M
00050      102 FORMAT('1','OUTPUT FROM PREBIC'//',',TIME ALIGNMENT INCORRECT,
00051      1 TIME AND DATE OF THE DATA INPUT MUST AGREE WITH THE SPECIFICA
00052      2TION OF NDATE AND NHR'//',',THE TIME ON THE
00053      3CARD WAS', 4(I3,''),',THE TIME REQUESTED WAS NHR,NDATE=', 4(I3,'')
00054      4,'THIS OCCURRED FOR CARD NO.',I3)
00055      STOP 2400
00056      C      ECHO DATA

```

```

00057      2 IF(M.EQ.1) WRITE(6,103)
00058 103 FORMAT('1',20X,'OUTPUT FROM PREBIC'//'*','CARD',T29,'
00059     IN BY SPECIES    ''   , NO.  , 'TYPE' , ' I J' , ' YR MO DA HR',2X,
00060     2 CO    NOX    NO2    NO    S02    PART CH4    NMHC    THC
00061     3 03')
00062 104 FORMAT('0',2I4,1X,6I3,2X, 5F7.2 , F7.0,4F7.2)
00063     WRITE(6,104) M,ITYPE,II(M),JJ(M),MDATE(3),MDATE(1),MDATE(2),
00064     1MHR,(C(L,M),L=1,NS)
00065     GO TO 1
00066 9001 FORMAT('1','OUTPUT FROM PREBIC'//'*','CODE STOPPED
00067     1BECAUSED CARD IMAGE TYPES NOT IN CORRECT ORDER.'//THE 1ST CARDS
00068     2MUST BE TYPE 31 FOLLOWED BY CARDS OF TYPE 32.'//THE CARD IMAGE
00069     3WHICH HALTED THE PROGRAM WAS TYPE= ',G4.2)
00070     900 WRITE(6,9001) ITYPE
00071     STOP 31
00072 C   ALL CONCENTRATIONS ARE LOADED, NOW COMPUTE INTERPOLATED VALUES
00073 C   AND BOUNDARY VALUES.
00074     3 NOSITE=M-1
00075 1041 FORMAT(///'*','TOTAL NUMBER OF AQ SITES FOR THIS HOUR IS:',1I2)
00076     WRITE(6,1041) NOSITE
00077 1042 FORMAT(///'T13,' MOUNTAIN BOUNDARIES OCCUR AT ://'
00078     2      3('0',17I3)//'T12,' RELATIVE TO THE INNER GRID')
00079     WRITE(6,1042) IBC
00080     DO 26 I=1,MS
00081     CMIN(I)=10000.
00082     26 CMAX(I)=0.
00083 C   REORGANIZE POLLUTANTS TO MADCAP SEQUENCE AND GRID
00084 C   MADCAP SEQUENCE
00085     C 1 CO 3NO2 5HC1 7HC3 15S02
00086     C 2 03 4NO 6HC2 8HC4
00087 C   CONVERT TO INNER GRID COORDINATES AND LOAD
00088 C   CONCENTRATIONS INTO MADCAP SEQUENCE
00089 DO 4 N=1,NOSITE
00090     IX(N)=II(N)-MIX
00091     JY(N)=JJ(N)-MIY
00092     C1(5,N)=0.44*C(8,N)
00093     IF(C1(5,N).LT.0.) C1(5,N)=-66.0
00094     IF(C1(5,N).LT.0.) GO TO 221
00095     CMIN(5)=AMIN1(CMIN(5),C1(5,N))
00096     CMAX(5)=AMAX1(CMAX(5),C1(5,N))
00097     221 C1(6,N)= 1.1*C(8,N)
00098     IF(C1(6,N).LT.0.0) C1(6,N)=-66.0
00099     IF(C1(6,N).LT.0.) GO TO 222
00100     CMIN(6)=AMIN1(CMIN(6),C1(6,N))
00101     CMAX(6)=AMAX1(CMAX(6),C1(6,N))
00102     222 IF(C(8,N).LT.0.0) C1(7,N) = -66.
00103     IF( C(8,N).LT.0.) GO TO 223
00104     C1(7,N) = 0.0
00105     CMIN(7)=AMIN1(CMIN(7),C1(7,N))
00106     CMAX(7)=AMAX1(CMAX(7),C1(7,N))
00107     223 IF(C(3,N).GE.0.) GO TO 21
00108     IF(C(4,N).GE.0.) GO TO 22
00109     GO TO 25
00110     22 IF(C(2,N).GE.0.) GO TO 23
00111     GO TO 25
00112

```

```

00113      23 C(3,N)=C(2,N)-C(4,N)
00114      IF(C(3,N).GE.0.) GO TO 25
00115      STOP 4
00116      21 IF(C(4,N).GE.0.) GO TO 25
00117      IF(C(2,N).GE.0.) GO TO 24
00118      GO TO 25
00119      24 C(4,N)=C(2,N)-C(3,N)
00120      IF(C(4,N).GE.0.) GO TO 25
00121      STOP 3
00122      25 CONTINUE
00123      IF(C1(1,N).LT.0.) GO TO 224
00124      CMAX(1)=AMAX1(CMAX(1),C1(1,N))
00125      CMIN(1)=AMIN1(CMIN(1),C1(1,N))
00126      224 CONTINUE
00127      IF(C1(3,N).LT.0.) GO TO 225
00128      CMAX(3)=AMAX1(CMAX(3),C1(3,N))
00129      CMIN(3)=AMIN1(CMIN(3),C1(3,N))
00130      225 CONTINUE
00131      IF(C1(4,N).LT.0.) GO TO 226
00132      CMAX(4)=AMAX1(CMAX(4),C1(4,N))
00133      CMIN(4)=AMIN1(CMIN(4),C1(4,N))
00134      226 C1(2,N)=C(10,N)
00135      IF(C1(2,N).LT.0.) GO TO 4
00136      CMAX(2)=AMAX1(CMAX(2),C1(2,N))
00137      CMIN(2)=AMIN1(CMIN(2),C1(2,N))
00138      4 CONTINUE
00139      11144 FORMAT('///' DATA ECHO ON INNER GRID BY MADCAP SPECIES SEQUENCE',
00140      1        4X,'(PPM)///
00141      2' NO.   I    J    CO    OZONE   NO2     NO     HC1     HC2     HC3
00142      3' )
00143      WRITE(6,11144)
00144      WRITE(6,11114) (N,IX(N),JY(N),(C1(L,N),L=1,MS),N=1,NOSITE)
00145      11114 FORMAT(' ',I2,1X,I3,1X,I3,1X,7(F6.2,2X)))
00146      11444 FORMAT(12X,7(1X,5(''),2X)/7X,'CMAX=',7(F6.2,2X)/7X,'CMIN=',7(F6.2
00147      2        ,2X))
00148      WRITE(6,11444) (CMAX(I),I = 1,7),(CMIN(I),I = 1,7)
00149      C BEGIN PROCESSING THIS HOUR'S CONCENTRATION
00150      C DATA YIELD, CE, BCX, BCY
00151      C LOOP OVER SPECIES(L=1 TO MS)
00152      C DO 12 L=1,MS
00153      C IF(L.LT.6.OR.L.GT.14) GO TO 6
00154      C ZERO CE,BCX,BCY FOR L=6,7,---14
00155      C DO 5 J=1,MY
00156      C BCX(L,J,1)=0.0
00157      C BCX(L,J,2)=0.0
00158      C DO 5 I=1,MX
00159      C CE(I,J,L)=0.0
00160      C IF(J.EQ.1) BCY(L,I,1)=0.0
00161      C IF(J.EQ.MY) BCY(L,I,2)=0.0
00162      C 5 CONTINUE
00163      C GO TO 12
00164      C LOOP OVER I-J CELLS
00165      6 DO 9 J=1,MY
00166      DO 9 I=1,MX
00167      DO 119 L=1,MS
00168      ANORM(L) = 0.0

```

```

00169      119 CON(L)=0.
00170      C   LOOP OVER SITES (1 TO NOSITE)
00171          DO 8 N=1,NOSITE
00172      C   CHECK FOR VALID DATA
00173          RSQ=(I-IX(N))**2+(J-JY(N))**2+1.E-30
00174          DO 8 L=1,MS
00175          IF(C1(L,N).LT.00.0) GO TO 8
00176          CON(L)=CON(L)+C1(L,N)/RSQ
00177          ANORM(L)=ANORM(L)+1./RSQ
00178      8 CONTINUE
00179          DO 9 L=1,MS
00180          9 CE(1,J,L)=CON(L)/ANORM(L)
00181      C   LOAD BCX AND BCY
00182          DO 12 L=1,MS
00183          DO 10 J=1,MY
00184          BCX(L,J,1)=CE(IC(J),J,L)
00185          10 BCX(L,J,2)=CE(IBC(J),J,L)
00186          DO 11 I=1,MX
00187          BCY(L,I,1)=CE(I,1,L)
00188          11 BCY(L,I,2)=CE(I,MY,L)
00189      12 CONTINUE
00190          MS1=MS-1
00191          DO 15 L=1,MS1
00192          WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),L,NAME(L)
00193          CALL OUTPUT(CE(1,1,L),MX,MY,MIX,MIY,CMAX(L),IBC)
00194          WRITE(6,108) NDATE(1),NDATE(2),NDATE(3),L,NAME(L)
00195          108 FORMAT('1',I2,'/',I2,'/')
00196          112,5X,'PLOT OF CE(I,J,',I1,') FROM PREBIC',5X,A4,'(PPMD')
00197          15 CALL PL(CE(1,1,L),MX,MY,CMAX(L),CMIN(L))
00198          107 FORMAT('1',I2,'/',I2,'/')
00199          112,5X,'PRINT OF CE(I,J,',I1,') FROM PREBIC:',3X,A4,'(PPMD')
00200      C
00201      C   FOR TEST PURPOSES PRINT B C X AND B C Y
00202      C
00203          703 FORMAT(1H1)
00204          WRITE(6,703)
00205          DO 40 L = 1,MS1
00206          WRITE(6,704) NAME(L)
00207          704 FORMAT('/1X,' B C X & B C Y FOR ',A4)
00208          WRITE(6,705) (BCY(L,I,2),I = 1,MX)
00209          705 FORMAT(7X,20F6.3)
00210          WRITE(6,706) (BCX(L,MY+1 - J,1),BCX(L,MY+1 - J,2),J = 1,MY)
00211          706 FORMAT((1X,F6.3,120X,F6.3))
00212          40 WRITE(6,705) (BCY(L,I,1),I = 1,MX)
00213      C   WRITE THIS HOUR'S OUTPUT TO TAPE 10
00214      C   FIRST RECORD IS THE HEADER
00215          WRITE(10) NDATE ,NHR
00216      C   SECOND RECORD IS CE
00217          WRITE(10) (((CE(I,J,K),I=1,MX),J=1,MY),K=1,7)
00218      C   THIRD RECORD IS BCX,BCY
00219          C   WRITE( ?) (((BCX(L,J,1),L=1,MS),J=1,MY),K=1,NZ),
00220          C   WRITE(10) (((BCX(L,J,1),L=1,7),J=1,MY),M=1,2),
00221          C   1(((BCY(L,I,1),L=1,7),I=1,MX),M=1,2)
00222      C   1(((BCX(L,J,2),L=1,MS),J=1,MY),K=1,NZ),
00223      C   2(((BCY(L,I,1),L=1,MS),I=1,MX),K=1,NZ),
00224      C   3(((BCY(L,I,2),L=1,MS),I=1,MX),K=1,NZ)
00225      C   STOP
00226      C   RETURN
00227      C   END

```

```

00001      SUBROUTINE OUTPUT(ARRAY,MX,MY,MIX,MIY,XMAX,IBC)
00002      C ..... OUTPUT IS CALLED TO PRINT ANY MX*MY ARRAY
00003      C ..... USING AN OFFSET OF MIX AND MIY AND GIVEN THE
00004      C ..... MAXIMUM VALUE IN THE ARRAY = XMAX (WHICH MUST BE POSITIVE)
00005      C
00006      C
00007      C
00008      C ..... PRINT THE EULERIAN CONCENTRATIONS
00009      C      DIMENSION ICE( 50 ),           ARRAY(MX,MY), IBC(MY)
00010      C
00011      C ..... FIND FACTOR FOR A TWO DIGIT PRINT .....
00012      C
00013      F = 1.
00014      FACTOR = 1.
00015      IF(XMAX.LT.1.E-30) GO TO 11
00016      LOGX = IFIX ALOG10(XMAX)
00017      IF(XMAX .LT. 1.0) LOGX = LOGX - 1
00018      FACTOR = 10.**(LOGX - 1)
00019      F = 1./FACTOR
00020      11 CONTINUE
00021      102 FORMAT('0',T47,'MULTIPLY VALUES IN THIS TABLE BY',1PE3.1)
00022      WRITE(6,102) FACTOR
00023      MOS=MIX+1
00024      MOT=MX+MIX
00025      103 FORMAT(' ',5X,42(I3))
00026      WRITE(6,103) (I,I=MOS,MOT)
00027      WRITE(6,103) (I,I = 1, MX)
00028      C
00029      C ..... PRINT CONCENTRATIONS .....
00030      C
00031      DO 30 JJ = 1,MY
00032      J = MY - JJ + 1
00033      JL=J+MIY
00034      LX = IBC(J)
00035      DO 15 I = 1,LX
00036      A = ARRAY(I,J)*F + 0.5
00037      IF(A.LT.0.5) A = A - 1.0
00038      15 ICE(I) = IFIX(A)
00039      104 FORMAT('0',I2,1X,I2,42(I3))
00040      WRITE(6,104) JL,J,(ICE(I),I = 1, LX)
00041      30 CONTINUE
00042      RETURN
00043      END

```

```

00001      C
00002      C      SUBROUTINE PL(Z,NX,NY,ZMAX,ZMIN)
00003      C*****
00004      C*
00005      C*          P L  IS A PRINTER-PLOTTER ROUTINE WHICH
00006      C*          SCALES MATRIX VALUES FROM 0 TO 9
00007      C*
00008      C*****
00009      C
00010      DIMENSION LQ(72),L(131),Z(NX,NY)
00011      DATA NNC,AMAX,AMIN,LOGFL/10,0.,0.,0/
00012      DATA LQ /'0','1','2','3','4','5',
00013      .   '6','7','8','9','A','B',
00014      .   'C','D','E','F','G','H',
00015      .   'J','K','L','M','N','O',
00016      .   'P','Q','R','S','T','U',
00017      .   'W','X','Y','Z','/,
00018      DATA ISTAR/*/,IMAX/60/
00019      C
00020      C
00021      C
00022      LOGFL = 0
00023      AMIN = 0.0
00024      AMAX = 0.0
00025      NNC = 10
00026      NXMX=NX
00027      C ASPECT = DY/DX
00028      ASPECT = 1.0
00029      JMAX = .6*NY/NX*IMAX*ASPECT + .5
00030      NC = MIN0(NNC,36)
00031      ZMIN = AMIN
00032      ZMAX = AMAX
00033      ANX = NX
00034      ANY = NY
00035      IF (ABS(ZMAX-ZMIN) .GT. 1.0E-5) GO TO 40
00036      ZMAX = Z(1,1)
00037      ZMIN = ZMAX
00038      DO 20 I=1,NX
00039      DO 20 J=1,NY
00040      ZMAX = AMAX1(ZMAX,Z(I,J))
00041      20 ZMIN = AMIN1(ZMIN,Z(I,J))
00042      IF (ABS(ZMAX-ZMIN) .LT. 1.0E-20) RETURN
00043      40 IF (LOGFL .EQ. 0) GO TO 80
00044      IF (ZMIN .GT. 0.) GO TO 60
00045      WRITE (6,220)
00046      RETURN
00047      60 EZMAX = ZMAX
00048      EZMIN = ZMIN
00049      ZMAX = ALOG10(ZMAX)
00050      ZMIN = ALOG10(ZMIN)
00051      80 SC = (NC*2.-1.E-5)/(ZMAX-ZMIN)
00052      DX = ANX/IMAX
00053      DY = ANY/JMAX
00054      Y = ANY+DY
00055      DO 180 JJ=1,JMAX
00056      J = JMAX -JJ +1

```

```

00057      Y = Y-DY
00058      M = Y + .5
00059      DM = AMAX1(Y - M + .5, 1.0E-10)
00060      IF(M.EQ.0) DM = 1.0
00061      IF(M.GE.NY) DM = 0.0
00062      X = 0.
00063      DO 160 I=1,IMAX
00064      X = X+DX
00065      IF (I*j .EQ. 1 .OR. I*j .EQ. IMAX*jMAX) GO TO 140
00066      IF (I .EQ. 1 .AND. J .EQ. JMAX .OR. J .EQ. 1 .AND. I .EQ. IMAX)
00067      GO TO 140
00068      N = X + .5
00069      DN = AMAX1(X - N + .5, 1.0E-10)
00070      IF(N.LE.0) DN = 1.0
00071      IF(N.GE.NX) DN = 0.0
00072      IF (LOGFL .NE. 0) GO TO 100
00073      DMM = AMAX1(1. - DM, 1.0E-10)
00074      DNN = AMAX1(1. - DN, 1.0E-10)
00075      IF(M.EQ.0) DMM = 0.0
00076      IF(M.LE.0) M=1
00077      IF(N.EQ.0) DNN = 0.0
00078      IF(N.LE.0) N=1
00079      NP1=N+1
00080      IF(NP1.GT.NX) NP1=NX
00081      MP1=M+1
00082      IF(MP1.GT.NY) MP1=NY
00083      C = DN*(DM*Z(NP1,MP1)+DMM*Z(NP1,M))+DNN*(DM*Z(N,MP1)+DMM*
00084      .Z(N,M))
00085      GO TO 120
00086      100 C = DN*(DM* ALOG10(Z(N+1,M+1))+(1.-DM)* ALOG10(Z(N,M))+(1.-DN)*
00087      .(DM* ALOG10(Z(N,M+1))+(1.-DM)* ALOG10(Z(N,M)))
00088      120 IND = 1+SC*(C-ZMIN)
00089      IF (IND .LE. 0) IND = 1
00090      IF (IND .GT. 2*NC) IND = 2*NC
00091      L(I) = LQ(IND)
00092      GO TO 160
00093      140 L(I) = ISTAR
00094      160 CONTINUE
00095
00096      C      INDEX CHANGE TO MAKE PRINTOUT IN MATRIX - LIKE FORM
00097      C
00098      180 WRITE (6,240) (L(I) , I = 1 , IMAX)
00099      IF (LOGFL .EQ. 0) GO TO 200
00100      ZMAX = HZMAX
00101      ZMIN = HZMIN
00102      200 WRITE (6,260) ZMIN,ZMAX
00103
00104      RETURN
00105      220 FORMAT (/////
00106      .          115H ****NON-POSITIVE VALUE IN ARRAY TO BE
00107      . CONTOURED LOGARITHMICALLY. PL IS RETURNING WITHOUT PLOTTING )
00108
00109      240 FORMAT (1X,131A1)
00110      260 FORMAT (/17H0MINIMUM VALUE = ,1PE9.3,5X,16HMAXIMUM VALUE = ,E9.3)
00111      280 FORMAT(1H1)
00112      END

```

### A.3.2 PREMET

```
00001 C THIS DRIVES PREMET UNDER THE SAME INPUT FILE STRUCTURE
00002 C ASSUMPTIONS AS WERE USED FOR PREBIC
00003 C
00004 C
00005 C DIMENSION NDATE(3), IBC1(60), IC1(60)
00006 C INTEGER DAY, YEAR, BEGIN
00007 C COMMON/GRIDA/MX, MY, MZ, MX, MIY
00008 C COMMON/TERRA/IBC1, IC1
00009 C NAMELIST /SETUP/ MONTH, DAY, YEAR, BEGIN, NOHRS
00010 C NAMELIST/GRID0/MX, MY, MZ, MX, MIY
00011 C NAMELIST/TERR0/IBC1, IC1
00012 C
00013 C
00014 C READ(5, SETUP)
00015 C WRITE(6, SETUP)
00016 C READ(5, GRID0)
00017 C WRITE(6, GRID0)
00018 C READ(5, TERR0)
00019 C WRITE(6, TERR0)
00020 C NDATE(1) = MONTH
00021 C NDATE(2) = DAY
00022 C NDATE(3) = YEAR
00023 C IHR = BEGIN
00024 2 READ(5, 101) ITYPE
00025 C IF( ITYPE.NE.72) GO TO 2
00026 C CALL PREMET(NDATE, IHR)
00027 C DO 10 N = 1, NOHRS
00028 1 READ(5, 101) ITYPE
00029 C IF( ITYPE.NE.32) GO TO 1
00030 C CALL PREMET(NDATE, IHR)
00031 C IHR = IHR + 1
00032 C IF( IHR.LT.24) GO TO 10
00033 C IHR = 0
00034 C NDATE(2) = NDATE(2) + 1
00035 10 CONTINUE
00036 C STOP
00037 101 FORMAT(12)
00038 C END
```

```

00001      SUBROUTINE PREMET(NDATE, NHR)
00002      C--NMET IS 1+MAXIMUM NUMBER OF ANTICIPATED METEOROLOGY CARDS(TYPE 32)
00003      C
00004      C--NINV IS 1+MAXIMUM NUMBER OF ANTICIPATED INVERSION CARDS(TYPE 41)
00005      C
00006      C-----NPIB IS 1+MAXIMUM NUMBER OF PIBAL SITES PER HOUR
00007      C
00008      C-----NPEL IS MAXIMUM NUMBER OF PIBAL ELEVATIONS ANTICIPATED
00009      C
00010      C
00011      C     COMMON   V(MX,MY,MZ) , U(MX,MY,MZ)
00012      C     DIMENSION  II(NMET) , JJ(NMET) , NDATE(3) , TEMP(NMET) ,
00013      C                   RELHUM(NMET) , WD(NMET) , CC(NMET) , CH(NMET) ,
00014      C                   UV(NMET) , NDATE(3) , IX(NMET) , JY(NMET) , IBC(MY) ,
00015      C
00016      C                   ZC(MZ) , IVAP(13) , IURB(13) ,
00017      C
00018      C                   LETTER(6) ,
00019      C
00020      C                   I41(NINV) , J41(NINV) , ZBOT(NINV) , TBOT(NINV) , ZTOP(NINV) ,
00021      C                   TTOP(NINV) , TGRAD(NINV) ,
00022      C                   P(7) , DMAX(MZ) , DMIN(MZ) ,
00023      C                   Z(5) , ZG(5) , WDV(NPEL) , WSV(NPEL) , IIX(NPIB) , JJY(NPIB)
00024      C                   WS(NMET, NZ) ,
00025      C                   Z0(NX,NY) , LND(NX,NY) , ZIBSE(MX,MY) , ZITOP(MX,MY) ,
00026      C                   KSTAB(MX,MY) , ISTB(8,6) , A(3,12) , B(3,12) ,
00027      C
00028      C                   SOL(MX,MY) , DEN(40) , H2O(MX,MY) ,
00029      C
00030      C                   TT(MX,MY) , DTDZ(MX,MY) , ZITOP(MX,MY) ,
00031      C                   PWS(NPIB,MZ) , PWD(NPIB,MZ)
00032      C     DIMENSION
00033      C     DIMENSION
00034      C     DIMENSION
00035      1    V(37,51,5) , U(37,51,5)
00036      2    II(50) , JJ(50) , MDATE(3) , TEMP(50) , IDATE(3) ,
00037      3    RELHUM( 50) , WD( 50) , CC( 50) , CH( 50) ,
00038      4    UV(50) , NDATE(3) , IX(50) , JY(50) , IBC(51) , IBC1(60) ,
00039      5    ZC( 5) , IVAP(13) , IURB(13) ,
00040      6    LETTER(6) ,
00041      7    I41( 10) , J41( 10) , ZBOT( 10) , TBOT( 10) , ZTOP( 10) ,
00042      8    TTOP( 10) , TGRAD( 10) ,
00043      1    P(7) , DMAX(5) , DMIN(5) ,
00044      2    ZP(25) , ZG(25) , WDV(25) , WSV(25) , IIX(25) , JJY(25)
00045      3    CLOUD(37,51) , CEIL(37,51) , T(37,51) , RH(37,51) ,
00046      4    WS( 50, 5) ,
00047      5    Z0(50,60) , LND(50,60) , ZIBSE(37,51) , ZITOP(37,51) ,
00048      6    KSTAB(37,51) , ISTB(8,6) , A(3,12) , B(3,12) ,
00049      7    SOL(37,51) , DEN(40) , H2O(37,51) ,
00050      8    TT(37,51) , DTDZ(37,51) ,
00051      9    PWS(25,5) , PWD(25,5)
00052      0    , NDCOD(2,2)
00053      1    DIMENSION IC1(60) , D(37,51, 5) , SPEED(37,51)
00054      2    REAL METDIR
00055      3    COMMON/TERRA/ IBC1 , IC1
00056      4    COMMON/GRIDA/MX,MY,MZ,MIX,MIY
00057      5    EQUIVALENCE (II,IX),(JJ,JY),(U(1,1,2),D(1,1,1)),SOL(1,1),
00058      6    TT(1,1),CLOUD(1,1)),

```

```

00057      2          (U(1,1,3),CEIL(1,1)),(U(1,1,4),T(1,1)),
00058      3          (U(1,1,5),RH(1,1),DTDZ(1,1),H2O(1,1))
00059      EQUIVALENCE (V(1,1,2),KSTAB(1,1))
00060      EQUIVALENCE (V(1,1,3),ZIBSE(1,1))
00061      EQUIVALENCE (V(1,1,4),ZITOP(1,1))
00062      EQUIVALENCE (V(1,1,5),SPEED(1,1))
00063 ****
00064 **** SITE DEPENDENT DATA ****
00065      DATA ALONG,ALAT,BRNG/117.083,32.75,120.0/, 
00066      1ZC/25.,87.5,187.5,375.,750./
00067 ****
00068      DATA ITME/0/
00069      IF( ITME.GT.0) GO TO 3471
00070      ITME= ITME+1
00071      NDCOD(1,2)=0
00072      NDCOD(2,2)=1
00073 C ****
00074 C      S E T      U P      D A T A
00075 C ****
00076 C      L O A D      Z0      A N D      L N D
00077 C ****
00078      72      READ(5,722) ITYPE, I,J1,J2,(Z0(I,J),J = J1,J2)
00079      IF( ITYPE.EQ.73) GO TO 2
00080      IF( ITYPE.NE.72) STOP 72
00081      722     FORMAT(I2,3I3,10F4.0)
00082      GO TO 72
00083      2      READ(5,733) ITYPE, I,J1,J2,(LND(I,J),J = J1,J2)
00084      IF( ITYPE.EQ.31) GO TO 7333
00085      733     FORMAT(I2,3I3,10I4)
00086      IF( ITYPE.NE.73) STOP 73
00087      GO TO 2
00088      7333     Z0MAX = -1.
00089      Z0MIN = 10000.
00090      LNDMAX = -1
00091      LNDMIN = 1000
00092      DO 73333 J = 1,60
00093      DO 73333 I = 1,50
00094      Z0MAX = AMAX1(Z0MAX,Z0(I,J))
00095      73333 Z0MIN = AMIN1(Z0MIN,Z0(I,J))
00096 C ****
00097 C      O U T P U T      Z0      A N D      L N D
00098 C ****
00099      WRITE(6,7221)
00100      7221     FORMAT(1H1,40X,'PRINT OF Z0(METERS) FOR SAN DIEGO COUNTY'//)
00101      CALL OUT3D(Z0,50,60,1,10)
00102 C      WRITE(6,7222)
00103      7222     FORMAT(1H1,16X,' PLOT OF Z0(METERS) FOR SAN DIEGO COUNTY'//)
00104 C      CALL PLC(Z0,50,60,Z0MAX,Z0MIN)
00105      WRITE(6,7331)
00106      7331     FORMAT(1H1,40X,'PRINT OF LAND USE FOR SAN DIEGO COUNTY'//)
00107      CALL OUT3D2(LND,50,60,1,10)
00108 C ****
00109 C      DATA Z0/3000*1.0/,LND/3000*1/
00110 C ****
00111 C . . . SHIFT MOUNTAIN BOUNDARIES TO THE INNER GRID . . . . . 853 C
00112 DO 27 NN = 1,MY

```



```

00169 CALL OUTPUT(CLOUD, MX, MY, MIX, MIY, CCMAX, IBC)
00170 WRITE(6, 107) NDCOD(1, 2), NDCOD(2, 2), MDATE, MHR
00171 CALL PL(CLOUD, MX, MY, CCMAX, CCMIN)
00172 CALL COAST(CH, NOSITE, MX, MY, IX, JY, CHMAX, CHMIN, 500., CEIL)
00173 WRITE(6, 108) NDCOD(1, 2), NDCOD(2, 2), MDATE, MHR
00174 CALL OUTPUT(CEIL, MX, MY, MIX, MIY, CHMAX, IBC)
00175 WRITE(6, 109) NDCOD(1, 2), NDCOD(2, 2), MDATE, MHR
00176 CALL PL(CEIL, MX, MY, CHMAX, CHMIN)
00177 CALL COAST(TEMP, NOSITE, MX, MY, IX, JY, TMAX, TMIN, 500., T)
00178 WRITE(6, 110) NDCOD(1, 2), NDCOD(2, 2), MDATE, MHR
00179 CALL OUTPUT(T, MX, MY, MIX, MIY, TMAX, IBC)
00180 WRITE(6, 111) NDCOD(1, 2), NDCOD(2, 2), MDATE, MHR
00181 CALL PL(T, MX, MY, TMAX, TMIN)
00182 CALL COAST(RELHUM, NOSITE, MX, MY, IX, JY, RHMAX, REMIN, 500., RH)
00183 WRITE(6, 112) NDCOD(1, 2), NDCOD(2, 2), MDATE, MHR
00184 CALL OUTPUT(RH, MX, MY, MIX, MIY, RHMAX, IBC)
00185 WRITE(6, 113) NDCOD(1, 2), NDCOD(2, 2), MDATE, MHR
00186 CALL PL(RH, MX, MY, RHMAX, REMIN)
00187 GO TO 14
00188 9001 WRITE(6, 9002) ITYPE
00189 STOP 32
00190 9002 FORMAT('1', 'OUTPUT FROM PREMET'//'*'//'*', 'CODE STOPPED
00191 1BECAUSE CARD IMAGE TYPES NOT IN CORRECT ORDER. THIS CARD
00192 2MUST BE CARD 32. THE CARD IMAGE WHICH HALTED THE PROGRAM WAS TYPE
00193 3= ', G10.3)
00194 C
00195 C      MESSAGE ABOUT WRONG TIME
00196 C
00197 8002 FORMAT('1', 'OUTPUT FROM PREMET'//'*', 'TIME ALIGNMENT INCORRECT'//
00198 1 THE TIME AND THE DATE OF DATA INPUT MUST AGREE WITH THE SPECIFICA
00199 2TION OF MDATE AND MHR IN THE DATA STATEMENT'//', 'THE TIME ON THE
00200 3CARD WAS MO/DA/YR/HR =',
00201 3      4(I3, ','), 'THE TIME REQUESTED WAS MO/DA/YR/HR', 4(I3, ','),
00202 4'THIS OCCURRED FOR CARD NO.', I3)
00203 106 FORMAT('1', 'OUTPUT FROM PREMET', 5X, 'VERSION=', I2, '//', I2
00204 1      //'*', I2, '//', I2, '//', I2.5X, 'PRINT OF CLOUD COVER', 10X
00205 2      , 'HR=', I2, '00PST')
00206 107 FORMAT('1', 'OUTPUT FROM PREMET', 5X, 'VERSION=', I2, '//', I2
00207 1      //'*', I2, '//', I2, '//', I2.5X, 'PLOT OF CLOUD COVER', 10X
00208 2      , 'HR=', I2, '00PST')
00209 108 FORMAT('1', 'OUTPUT FROM PREMET', 5X, 'VERSION=', I2, '//', I2
00210 1      //'*', I2, '//', I2, '//', I2.5X, 'PRINT OF CEILING HEIGHTS',
00211 1      2X, '(METERS WRT GROUND)', 5X
00212 2      , 'HR=', I2, '00PST')
00213 109 FORMAT('1', 'OUTPUT FROM PREMET', 5X, 'VERSION=', I2, '//', I2
00214 1      //'*', I2, '//', I2, '//', I2.5X, 'PLOT OF CEILING HEIGHTS',
00215 1      2X, '(METERS WRT GROUND)', 5X
00216 2      , 'HR=', I2, '00PST')
00217 103 FORMAT('1', 'OUTPUT FROM PREMET', 5X, 'VERSION=', I2, '//', I2
00218 1      //'*', I2, '//', I2, '//', I2.5X, 'GROUND
00219 1STATION METEOROLOGICAL DATA ECHO', 5X, 'HR=', I2, '00PST'
00220 1      /T14, 43(*')/
00221 2 J TEMP(C) RELHU(%) WD(DEG) WS(M/S) CC(%) CH(MD) PC(AT
00222 3M) UV(WATTS/MSQ)
00223 104 FORMAT('0', 2X, I2, 4X, I2, 1X, I2, 2X, F4.0, 5X, F5.0, 5X, F4.0, 5X, F5.1, 4X
00224 1, F4.0, 6X, F5.0, 5X, F6.0, 5X, F6.0)

```







```

00393 C THE RADIATION INDEX DEPENDS UPON THE ANGLE OF THE SUN ABOVE THE
00394 C SOUTHERN HORIZON, THE CLOUD COVER, CEILING HEIGHT, AND WHETHER
00395 C IT IS NIGHT OR DAY. SINCE ONLY THE SURFACE LAYER WIND SPEED IS
00396 C USED, THE PASQUILL STABILITY CLASS FOUND IS ONLY APPLICABLE
00397 C TO THE SURFACE LAYER. IF THERE IS AN URBAN HEAT ISLAND, THE
00398 C STABILITY CLASS IS DECREASED BY ONE (MADE MORE UNSTABLE).
00399 C

00400 ICLAS=4
00401 IF( SKY .GT. 0.9 .AND. CEIL(I,J).LT.2134.) GOTO 120
00402 IF((SNRIZ+1.).LT.MHR.AND.MHR.LT.(SNSET-1.)) GOTO 100
00403 IRAD=6
00404 IF( SKY .GT. 0.4) IRAD=5
00405 GOTO 1101
00406 100 IRAD=1
00407 IF(ALPHA.LE.60.) IRAD=2
00408 IF(ALPHA.LE.35.) IRAD=3
00409 IF(ALPHA.LE.15.) IRAD=4
00410 IF( SKY .LT. 0.5) GOTO 1101
00411 IF(CEIL(I ,J ).LT.4877.) IRAD=IRAD+1
00412 IF(CEIL(I ,J ).LT.2134.) IRAD=IRAD+1
00413 IF( SKY .GT. 0.9) IRAD=IRAD+1
00414 IF( IRAD.GT. 4) IRAD=4
00415 1101 IWND= SPEED(I,J)+1.5
00416 IF( IWND.GT.8) IWND=8
00417 ICLAS = ISTB( IWND, IRAD)
00418 120 IF( IURB(LND(I+MIX,J+MIY)).EQ. 1) ICLAS = ICLAS -1
00419 IF( ICLAS.LT.1) ICLAS=1
00420 KSTAB(I,J) = ICLAS
00421 C
00422 C
00423 C
00424 C ASSIGN SOLAR INSOLATION TO EVERY I-J ACCORDING TO CLOUD COVER
00425 C
00426 C---- CHECK FOR SUNRISE FLAG
00427 C
00428 IF( ISOL.LT. 1) GO TO 39
00429 IF( NOUV.GE.2) GO TO 36
00430 IF( NOUV.EQ.0) GO TO 37
00431 SOL(I,J)=UVMAX-S1*AMIN1(CC1, SKY )-S2*AMIN1(1.-CC1,DIM( SKY ,CC1))
00432 GO TO 38
00433 36 SOL(I,J)=UVMAX-S1*AMIN1(CC1, SKY )-S2*AMIN1(CC2-CC1,DIM( SKY ,CC1))
00434 1-S3*AMIN1(1.-CC2,DIM( SKY ,CC2))
00435 GO TO 38
00436 37 SOL(I,J)=UVMAX-(UVMAX-UVMIN)*SKY
00437 38 IF(SOL(I,J).LT.UVMIN) SOL(I,J)=UVMIN
00438 GO TO 5739
00439 39 SOL(I,J)=0.0
00440 5739 SOLMAX=AMAX1(SOLMAX,SOL(I,J))
00441 SOLMIN=AMIN1(SOLMIN,SOL(I,J))
00442 C
00443 C WATER VAPOR ASSIGNMENT
00444 C
00445 DATA DEN/4.847,5.192,5.559,5.947,6.36,6.797,7.26,7.75,8.27,8.619
00446 1,9.399,10.01,10.66,11.35,12.07,12.83,13.63,14.84,15.37,16.21,17.3,
00447 218.34,19.43,20.58,21.78,23.05,24.38,25.78,27.24,28.78,30.38,32.07,
00448 333.83,35.68,37.61,39.63,41.75,43.96,46.26,48.67/

```



```

00505      118 FORMAT('1','OUTPUT FROM PREMET' , 5X,'VERSION=' , I2,'/' , I2
00506      1      '/0', I2,'/' , I2,'/' , I2.5X,' PLOT OF WATER CONCENTRATION',
00507      1      2X, '(PPM)' , 5X
00508      2      , 'HR=' , I2, '00PST')
00509      CALL PL(H2O,MX,MY,H2OMAX,H2OMIN)
00510
00511 C-----WRITE TITLE LINE FOR SOL
00512 C
00513      WRITE(6,119) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00514      119 FORMAT('1','OUTPUT FROM PREMET' , 5X,'VERSION=' , I2,'/' , I2
00515      1      '/0', I2,'/' , I2,'/' , I2.5X,'PRINT OF SOLAR INTENSITY AS INFERR
00516      2ED FROM CLOUD COVER (WATTS/M**2)' , 5X
00517      2      , 'HR=' , I2, '00PST')
00518      CALL OUTPUT(SOL,MX,MY,MIX,MIY,SOLMAX,IBC)
00519      WRITE(6,1201) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00520      1201 FORMAT('1','OUTPUT FROM PREMET' , 5X,'VERSION=' , I2,'/' , I2
00521      1      '/0', I2,'/' , I2,'/' , I2.5X,' PLOT OF SOLAR INTENSITY AS INFERR
00522      2ED FROM CLOUD COVER (WATTS/M**2)' , 5X
00523      2      , 'HR=' , I2, '00PST')
00524      CALL PL(SOL,MX,MY,SOLMAX,SOLMIN)
00525
00526 C***** ****
00527 C***** ****
00528 C-----WRITE SOL & WATER ON TAPE 11
00529 C***** ****
00530 C-----HEADER
00531      WRITE(11) MDATE , MHR
00532
00533 C-----H2O AND SOL RECORD
00534 C
00535      WRITE(11) ((SOL(I,J) , I=1,MX) , (H2O(I,J) , I=1,MX) , J=1,MY)
00536
00537 C-----READ CARD 41 :::: INVERSI ONS ::::
00538 C
00539 8888 CONTINUE
00540      N=0
00541      41 N=N+1
00542      READ(5,121) ITYPE,I41(N),J41(N),IDATE(3),IDATE(1),IDATE(2),IHR,
00543      1Z41,ZBOT(N),TBOT(N),RH1,ZTOP(N),TTOP(N),RH2
00544      121 FORMAT(I2,2I3,4I2, 2F4.0,F2.0 ,F4.0,12X, F4.0,F2.0,F4.0)
00545
00546 C-----CHECK CARD TYPE
00547 C
00548      IF( ITYPE.EQ.81) GO TO 126
00549      IF( ITYPE.NE.41.AND. ITYPE.NE.42) GO TO 9007
00550      IF( ITYPE.EQ.42) GO TO 126
00551      MDATE(1)=IDATE(1)
00552      MDATE(2)=IDATE(2)
00553      MDATE(3)=IDATE(3)
00554      MHR= IHR
00555      IF( MDATE(1).EQ. NDATE(1).AND. MDATE(2).EQ. NDATE(2).AND. MDATE(3).EQ.
00556      1NDATE(3).AND. MHR.EQ. NHR) GO TO 123
00557      WRITE(6,8002) MDATE(1),MDATE(2),MDATE(3),MHR
00558      STOP 24
00559      123 ZTOP41=ZTOP(N)
00560      ZTOP(N)=ZTOP41-Z41

```

```

00561      ZBOT41=ZBOT(N)
00562      ZBOT(N)=ZBOT41-Z41
00563      C
00564      C-----ECHO DATA
00565      C
00566      IF(N .EQ. 1) WRITE(6,124) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00567      124 FORMAT('1','OUTPUT FROM PREMET', 5X,'VERSION=',I2,'/',I2
00568      1      '/0',I2,'/',I2,'/',I2,T24,'ECHO OF INVERSION DATA',5X
00569      2      , 'HR=',I2,'00PST/T24,22(*')
00570      1/T8,'LOCATION',T29,'BOTTOM OF INVERSION',T65,'TOP OF INVERSION'
00571      2/T8,8(''),T25,27(''),T61,24('')/
00572      32X,'SITE ',T10,' I J',T13,'MSL(M)',T25,'(MD WRTG',T37,'(MD MSL',
00573      4T48,'TEMP(C)',T59,'(MD WRTG',T71,'(MD MSL',T82,'TEMP(C)')
00574      WRITE(6,125) N,I41(N),J41(N),Z41,ZBOT(N),ZBOT41,TBOT(N),ZTOP(N)
00575      1,ZTOP41,TTOP(N)
00576      125 FORMAT(/2X,I2,T9,I2,1X,I2,T19,F5.0,T26,F5.0,T38,F5.0,T49,F5.1,T60,
00577      1F5.0,T72,F5.0,T83,F5.1)
00578      GO TO 41
00579      C
00580      C-----SHIFT TO INNER GRID
00581      126 N41=N-1
00582      IF(N41.LT.1.OR.N41.GT.50) GO TO 9005
00583      C     SHIFT TO INNER GRID
00584      DO 127 N=1,N41
00585      I41(N)=I41(N)-MIX
00586      J41(N)=J41(N)-MIY
00587      127 CONTINUE
00588      C
00589      C-----ADJUST INVERSION DATA TO AGREE WITH GROUND TEMP
00590      C
00591      DATA STABLE/.015/,GAMMA/.0098/
00592      DO 128 N=1,N41
00593      IN=I41(N)
00594      JN=J41(N)
00595      C
00596      C . . . CHECK FOR DATUM OUTSIDE INNER GRID . . . . .
00597      C
00598      IF(IN.LT.1) IN = 1
00599      IF(IN.GT.MX) IN = MX
00600      IF(JN.LT.1) JN = 1
00601      IF(JN.GT.MY) JN = MY
00602      C     NOW CATCH THE CASE OF NO INVERSION
00603      C     I.E. ---THE LACK OF AN INVERSION IS DENOTED IN THE
00604      C     DATA BY ZBOT=ZTOP=DATA INFERENCE VALUE OF ZTOP
00605      C     NOTE: IN THIS CODE A FINITE INVERSION TOP HEIGHT IS ALWAYS ASSUMED
00606      C
00607      IF(ABS(ZBOT(N)-ZTOP(N)).LT.0.01) GO TO 130
00608      TGRAD(N)=(TTOP(N)-TBOT(N))/(ZTOP(N)-ZBOT(N))
00609      C
00610      C     TEST FOR THE CASE OF A GROUND-BASED INVERSION
00611      C
00612      IF(ZBOT(N).LT.1.) GO TO 129
00613      TP=T(IN,JN)-ZBOT(N)*GAMMA
00614      TTOP(N)=TTOP(N)+TP-TBOT(N)
00615      GO TO 128
00616      129 TTOP(N)=TTOP(N)+T(IN,JN)-TBOT(N)

```

```

00617      GO TO      128
00618      130 TGRAD(N)=STABLE
00619      TTOP(N)=T(IN,JN)-GAMMA*ZTOP(N)
00620      128 CONTINUE
00621      GO TO 132
00622      9005 WRITE(6,9006) N41
00623      9006 FORMAT('1','OUTPUT FROM PREMET'// ' N41 SHOULD NOT BE GREATER THAN
00624      150 OR LESS THAN 1 N41= ',G10.2)
00625      STOP 41
00626      9007 WRITE(6,9008) ITYPE
00627      STOP 41
00628      9008 FORMAT('1','OUTPUT FROM PREMET'/1X ,18('*')//'*','CODE STOPPED
00629      1 BECAUSE CARD IMAGE TYPES NOT IN CORRECT ORDER. THIS CARD
00630      2 MUST BE CARD 41. THE CARD IMAGE WHICH HALTED THE PROGRAM WAS
00631      3 TYPE = ', G10.3 )
00632      C
00633      C----- SIMPLIFIED TREATMENT IF ONLY ONE INVERSION DATUM EXISTS
00634      C
00635      132 IF(N41.EQ.1) GO TO 133
00636      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
00637      IAM=44
00638      WRITE(6,70001) IAM,TTOP
00639      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
00640      CALL COAST(TTOP,N41,MX,MY,I41,J41,TMAX,TMIN,100.,TT)
00641      CALL COAST(TGRAD,N41,MX,MY,I41,J41,TGMAX,TGMIN,100.,DTDZ)
00642      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
00643      IAM = 6
00644      WRITE(6,70001) IAM,N41,MX,MY, TOPMAX, TOPMIN
00645      WRITE(6,70001) IAM,(ZTOP(N),I41(N),J41(N),N=1,3)
00646      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
00647      CALL COAST(ZTOP,N41,MX,MY,I41,J41, TOPMAX, TOPMIN,100.,ZITOP)
00648      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
00649      IAM = 7
00650      WRITE(6,70001) IAM,N41,MX,MY, TOPMAX, TOPMIN
00651      WRITE(6,70001) IAM,(ZTOP(N),I41(N),J41(N),N=1,3)
00652      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
00653      C --- PRINT AND PLOT OF TT, DTDZ, ZITOP
00654      C
00655      WRITE(6,9205) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00656      9205 FORMAT('1','OUTPUT FROM PREMET' , 5X,'VERSION=' , I2,'/' , I2
00657      1      '/0', I2,'/' , I2,'/' , I2,5X,'PRINT OF TEMPERATURE OF INVERSION
00658      2TOP (C) CONSISTENT WITH GROUND TEMPERATURES' , 5X
00659      2      , 'HR=' , I2,'00PST')
00660      CALL OUTPUT(TT,MX,MY,MIX,MIY,TMAX,IBC)
00661      WRITE(6,9206) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00662      9206 FORMAT('1','OUTPUT FROM PREMET' , 5X,'VERSION=' , I2,'/' , I2
00663      1      '/0', I2,'/' , I2,'/' , I2,5X,'PLOT OF TEMPERATURE OF INVERSION
00664      2TOP (C)' , 5X
00665      2      , 'HR=' , I2,'00PST')
00666      CALL PL(TT,MX,MY,TMAX,TMIN)
00667      WRITE(6,9207) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00668      9207 FORMAT('1','OUTPUT FROM PREMET' , 5X,'VERSION=' , I2,'/' , I2
00669      1      '/0', I2,'/' , I2,'/' , I2,5X,'PRINT OF TEMPERATURE GRADIENT OF
00670      2THE INVERSION LAYER (DEGREES C/M' , 5X
00671      2      , 'HR=' , I2,'00PST')
00672      CALL OUTPUT(DTDZ,MX,MY,MIX,MIY,TGMAX,IBC)

```

```

00673      WRITE(6,9208) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00674      9208 FORMAT('1','OUTPUT FROM PREMET' , 5X,'VERSION= ',I2,'/',I2
00675      1      '/0',I2,'/',I2,'/',I2,5X,'PLOT OF TEMPERATURE GRADIENT OF
00676      2THE INVERSION LAYER (DEGREES C/M)',5X
00677      2      ,HR= ',I2,'00PST')
00678      CALL PL(DTDZ,MX,MY,TGMAX,TGMIN)
00679      WRITE(6,9209) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00680      9209 FORMAT('1','OUTPUT FROM PREMET' , 5X,'VERSION= ',I2,'/',I2
00681      1      '/0',I2,'/',I2,'/',I2,5X,'PRINT OF TOP OF INVERSION LAYER
00682      2(METERS WRT GROUND)',5X
00683      2      ,HR= ',I2,'00PST')
00684      CALL OUTPUT(ZITOP,MX,MY,MIX,MIY,TOPMAX,IBC)
00685      WRITE(6,9210) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00686      9210 FORMAT('1','OUTPUT FROM PREMET' , 5X,'VERSION= ',I2,'/',I2
00687      1      '/0',I2,'/',I2,'/',I2,5X,'PLOT OF TOP OF INVERSION LAYER
00688      2(METERS WRT GROUND)',5X
00689      2      ,HR= ',I2,'00PST')
00690      CALL PL(ZITOP,MX,MY,TOPMAX,TOPMIN)
00691      C-----LOOP OVER CELLS
00692      C
00693      BOTMAX=0.
00694      BOTMIN=1000.
00695      DO 131 I=1, MX
00696      DO 131 J=1, MY
00697      TDIFF=TTC(I,J)-T(I,J)
00698      TOP=ZITOP(I,J)*DTDZ(I,J)-TDIFF
00699      ZIBSE(I,J)=TOP/(DTDZ(I,J)+GAMMA)
00700      IF(ZIBSE(I,J).LT.1.) GO TO 140
00701      ZIBSEP=ZIBSE(I,J)+10.
00702      IF(ZIBSEP.GT.ZITOP(I,J))          GO TO 141
00703      GO TO 1310
00704      C
00705      C      IF THERE IS NO INVERSION AT (I,J) THEN BASE=TOP=-60
00706      C
00707      141 ZITOP(I,J)=-60.
00708      ZIBSE(I,J)=-60.
00709      GO TO 131
00710      C
00711      C      GROUND BASED INVERSION AT THIS I-J
00712      C
00713      140 ZIBSE(I,J)=0.
00714      1310 BOTMAX=AMAX1(BOTMAX,ZIBSE(I,J))
00715      BOTMIN=AMIN1(BOTMIN,ZIBSE(I,J))
00716      131 CONTINUE
00717      GO TO 150
00718      133 TOPMAX = ZTOP(1)
00719      TOPMIN = 0.0
00720      C
00721      C      CATCH THE CASE OF ONE DATUM
00722      C      REPRESENTING A WHOLLY NEUTRAL ATMOSPHERE
00723      C
00724      IF(TGRAD(1).EQ.STABLE) GO TO 139
00725      R= 1./(TGRAD(1)+GAMMA)
00726      S=(TGRAD(1)*ZTOP(1)-TTOP(1))*R
00727      BOTMAX=0.
00728      BOTMIN =1000.

```

```

00729      DO 134 J=1,MY
00730      DO 134 I=1,MX
00731      ZITOP(I,J)=ZTOP(1)
00732      ZIBSE(I,J)=S+T(I,J)*R
00733      IF(ZIBSE(I,J).LT.1) GO TO 138
00734      ZIBSEP=ZIBSE(I,J)+10.
00735      IF(ZIBSEP.GT.ZITOP(I,J))      GO TO 137
00736      GO TO 1340
00737      137 ZIBSE(I,J)=-60.
00738      ZITOP(I,J)=-60.
00739      GO TO 134
00740      138 ZIBSE(I,J)=0.
00741      1340 BOTMAX=AMAX1(BOTMAX,ZIBSE(I,J))
00742      BOTMIN=AMIN1(BOTMIN,ZIBSE(I,J))
00743      134 CONTINUE
00744      GO TO 150
00745      139 DO 1391 I=1,MX
00746      DO 1391 J=1,MY
00747      ZIBSE(I,J)=-60.
00748      ZITOP(I,J)=-60.
00749      C-----PRINT ZITOP
00750      C
00751      1391 CONTINUE
00752      WRITE(6,157)
00753      157   FORMAT('1',10X,'NO PRINT OR PLOT OF INVERSION ENTIRE REGION
00754      1 NEUTRAL')
00755      GO TO 188
00756      150 WRITE(6,151) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00757      151 FORMAT('1','OUTPUT FROM PREMET',5X,'VERSION=',I2,'/',I2
00758      1     '/0',I2,'/',I2,'/',I2,5X,'PRINT OF INVERSION TOP (MD AFTER
00759      2 CALCULATION OF INVERSION BASE',5X
00760      2     ',HR=',I2,'00PST')
00761      CALL OUTPUT(ZITOP,MX,MY,MIX,MIY,TOPMAX,IBC)
00762      C
00763      C-----PLOT OF ZITOP
00764      C
00765      WRITE(6,152) NDCOD(1,2),NDCOD(2,2),MDATE,MHR
00766      152 FORMAT('1','OUTPUT FROM PREMET',5X,'VERSION=',I2,'/',I2
00767      1     '/0',I2,'/',I2,'/',I2,5X,'PLOT OF INVERSION TOP (MD AFTER
00768      2 CALCULATION OF INVERSION BASE',5X
00769      2     ',HR=',I2,'00PST')
00770      161 FORMAT('1','OUTPUT FROM PREMET',5X,'VERSION=',I2,'/',I2
00771      1     '/0',I2,'/',I2,'/',I2,5X,'PLOT OF INVERSION TOP (MD AFTER
00772      2 CALCULATION OF INVERSION BASE',5X
00773      2     ',HR=',I2,'00PST')
00774      CALL PL(ZITOP,MX,MY,TOPMAX,TOPMIN)
00775      1610 WRITE(6,171) MDATE(1),MDATE(2),MDATE(3),NDCOD(1,2),NDCOD(2,2)
00776      171 FORMAT('1',20X,'OUTPUT FROM PREMET'/1X,I2,'/',I2,'/',I2,10X,'PRINT
00777      1 OF INVERSION BASE(METERS)',40X,'VERSION=',214)
00778      CALL OUTPUT(ZIBSE,MX,MY,MIX,MIY,BOTMAX,IBC)
00779      WRITE(6,181) MDATE(1),MDATE(2),MDATE(3),NDCOD(1,2),NDCOD(2,2)
00780      181 FORMAT('1',20X,'OUTPUT FROM PREMET'/1X,I2,'/',I2,'/',I2,10X,'PLOT
00781      1 OF INVERSION BASE(METERS)',40X,'VERSION=',214)
00782      CALL PL(ZIBSE,MX,MY,BOTMAX,BOTMIN)
00783      188 CONTINUE
00784      C      STOP 9999

```

00785 C  
 00786 C IVAP CONTAINS THE EVAPORATIVE POTENTIAL (1=LOW, 2=Moderate,  
 00787 C 3=HIGH) OF EACH OF THE 13 LAND USES.  
 00788 C  
 00789 C IURB CONTAINS THE URBANIZATION TYPE (1=URBAN, 2=RURAL) OF  
 00790 C EACH OF THE 13 LAND USES. IF A LAND USE IS URBAN, IT IS  
 00791 C CONSIDERED TO HAVE AN URBAN HEAT ISLAND AND TO INCREASE THE  
 00792 C INSTABILITY OF THE ATMOSPHERE BY ONE PASQUILL CLASS.  
 00793 C  
 00794 C ISTB CONTAINS THE TABLE OF PASQUILL STABILITY CLASSES, AS A  
 00795 C FUNCTION OF WIND SPEED VS RADIATION INDICES - DESCRIBED BY D.  
 00796 C BRUCE TURNER IN HIS "WORKBOOK FOR ATMOSPHERIC DISPERSION  
 00797 C ESTIMATES"  
 00798 C  
 00799 C A AND B CONTAIN THE CURVE COEFFICIENTS FOR THE Z0 VS THE  
 00800 C INVERSE OF THE MONIN-OBUKHOV LENGTH AS PRESENTED BY MYRUP &  
 00801 C RANZIERI. EACH OF THE 6 PASQUILL STABILITY CLASSES HAS A  
 00802 C SET OF 3 CURVES (FOR THE EVAPORATIVE POTENTIALS), AND EACH  
 00803 C OF THESE CURVES IS BROKEN INTO TWO PARTS AT Z0=0.2 METERS TO  
 00804 C FACILITATE CURVE-FITTING.  
 00805 C  
 00806 DATA IVAP/2,2,1,1,1,1,3,3,2,2,3,3,3/  
 00807 DATA IURB/1,1,1,1,1,1,2,2,2,2,2,2/  
 00808 DATA ISTB/1,1,1,2,2,3,3,3,  
 00809 1 1,2,2,2,3,3,3,4.  
 00810 2 2,2,3,3,3,4,4,4,  
 00811 3 3,3,4,4,4,4,4,4,  
 00812 4 6,6,5,5,4,4,4,4,  
 00813 5 6,6,6,6,5,5,4,4/  
 00814 DATA A/- .0897694, -.0897694, -.0611057,  
 00815 1 -.0897694, -.0897694, -.0611057,  
 00816 2 -.0611057, -.0352282, -.0079174,  
 00817 3 -.0611057, -.0352282, -.0128038,  
 00818 4 -.0079174, +.0013377, +.0090530,  
 00819 5 -.0128038, -.0065255, -.0013616,  
 00820 6 +.0090530, +.0000000, -.0109132,  
 00821 7 -.0013616, +.0000000, +.0015664,  
 00822 8 -.0109132, -.0072292, +.0162133,  
 00823 9 +.0015664, +.0064349, +.0105284,  
 00824 A +.0162133, +.0193795, +.0193795,  
 00825 B +.0105284, +.0193795, +.0193795/  
 00826 DATA B/+ .0377771, +.0377771, +.0336882,  
 00827 1 +.0377771, +.0377771, +.0336882,  
 00828 2 +.0336882, +.0290271, +.0261284,  
 00829 3 +.0336882, +.0290271, +.0176821,  
 00830 4 +.0261284, +.0206750, +.0146393,  
 00831 5 +.0176821, +.0098969, +.0044900,  
 00832 6 +.0146393, +.0000000, -.0151783,  
 00833 7 +.0044900, +.0000000, -.0027917,  
 00834 8 -.0151783, -.0263189, -.7496910,  
 00835 9 -.0027917, -.0105847, -.317610,  
 00836 A -.7496910, -.383930, -.383930,  
 00837 B -.317610, -.383930, -.383930/  
 00838 C  
 00839 C THE FOLLOWING ARE PHYSICAL CONSTANTS PI AND VON KARMON'S CNST.  
 00840 DATA PI/3.1415926/, VKAR/0.35/

```

00841 C
00842 C LONGITUDE AND LATITUDE OF THE STUDY AREA ARE STORED IN THE
00843 C FOLLOWING DATA STATEMENT. BRNG IS USED TO INDICATE THE TIME
00844 C ZONE - 120 FOR PST, 105 FOR MST, 90 FOR CST, OR 75 FOR EST.
00845 C LONG & LAT CURRENTLY STORED ARE THOSE FOR SAN DIEGO, AND BRNG
00846 C IS FOR PST.
00847 C
00848 C ZW IS THE HEIGHT ABOVE THE AVERAGE CANOPY HEIGHT AT WHICH THE
00849 C WIND IS MEASURED (OR CALCULATED), AND IS ASSUMED = 10 METERS.
00850 C DATA ZW/10./
00851 C
00852 C
00853 C
00854 C
00855 C
00856 C-----DIFFUSION(CALC. MONIN-OBUKHOV LENGTH)
00857 C-----SET DMAX AND DMIN
00858 DO 10125 K=1,MZ
00859 DMAX(K)=0.0
00860 10125 DMIN(K)=1000.
00861 DO 10290 I =1, MX
00862 DO 10290 J =1, MY
00863 C
00864 C MONIN-OBUKHOV LENGTH ROUTINE
00865 C
00866 C THE MONIN-OBUKHOV LENGTH (ZLEN) IS FOUND ACCORDING TO THE CURVES
00867 C PRESENTED BY MYRUP AND RANZIERI IN THEIR PAPER OF VERTICAL
00868 C DIFFUSIVITIES. IF A GIVEN Z0 EXCEEDS THE LIMITS OF THE CURVES
00869 C AS DRAWN IN THEIR PAPER, IT IS RESET TO THE LAST ACCEPTABLE
00870 C Z0 FOR THE STABILITY CLASS AND EVAPORATIVE POTENTIAL.
00871 C FOR NEUTRAL OR NEAR-NEUTRAL STABILITY CONDITIONS, THE
00872 C MONIN-OBUKHOV LENGTH IS ASSUMED TO HAVE A LARGE, NEGATIVE
00873 C VALUE (-1.E+05) AS AN APPROXIMATION OF INFINITY.
00874 C
00875 C Z0 IS NOT ALLOWED TO BE LESS THAN 0.02 METERS, TO PREVENT
00876 C UNDERFLOW/OVERFLOW PROBLEMS ON THE IBM 370/168 IN THE VERTICAL
00877 C DIFFUSIVITY ROUTINE. THE ENTIRE PROGRAM MUST BE DOUBLE PRECISION
00878 C IF IT IS NECESSARY TO HAVE Z0 LESS THAN 0.02 METERS.
00879 C
00880 ICLAS=KSTAB( I, J)
00881 ZOV=Z0( I+MIX, J+MIY)
00882 IF(ZOV.LT.0.02) ZOV=0.02
00883 IVP=IVAP(LND( I+MIX, J+MIY))
00884 M=1+( ICLAS-1)*2
00885 IF(ZOV.GE.0.2) M=M+1
00886 GOTO( 10130, 10140, 10150, 10160, 10170), ICLAS
00887 10130 IF( IVP.LT.3.AND.ZOV.LT.0.05) ZOV=0.05
00888 IF( IVP.EQ.3.AND.ZOV.LT.0.005) ZOV=0.005
00889 GOTO 10180.
00890 10140 IF( IVP.EQ.1.AND.ZOV.LT.0.005) ZOV=0.005
00891 GOTO 10180
00892 10145 IF( IVP.EQ.3.AND.ZOV.GE.1.0) GOTO 10210
00893 GOTO 10180
00894 10150 IF( IVP.EQ.2.OR.ZOV.GE.1.0) GOTO 10210
00895 GOTO 10180
00896 10160 IF( IVP.EQ.1.AND.ZOV.GE.1.0) GOTO 10210

```

```

00897      IF( IVP.EQ.1.AND.ZOV.LT.0.0035) ZOV=0.0035
00898      IF( IVP.EQ.3) GOTO 10190
00899      GOTO 10180
00900 10170  IF( IVP.GT.1.AND.ZOV.LT.0.1) ZOV=0.1
00901      IF( IVP.EQ.1.AND.ZOV.LT.0.0035) ZOV=0.0035
00902      GOTO 10190
00903 10180  ZLEN=A( IVP,MD+B( IVP,MD*ALOG10(ZOV)
00904      GOTO 10200
00905 10190  ZLEN=A( IVP,MD*EXP(B( IVP,MD)*ALOG10(ZOV))
00906 10200  IF(ABS(ZLEN).GE.0.001) GOTO 10220
00907 10210  ZLEN=-1.E-05
00908 10220  ZLEN=1./ZLEN
00909      C
00910      C
00911      C      KZ (VERTICAL DIFFUSIVITY) ROUTINE
00912      C
00913      C      THE ESSENCE OF THE MYRUP-RANZIERI VERTICAL DIFFUSIVITY MODEL
00914      C      IS CONTAINED IN THIS ROUTINE. ALSO CONTAINED IN THE ROUTINE
00915      C      IS THE Z-LOOP FOR BRINGING IN THE VARIOUS HEIGHTS FOR WHICH
00916      C      THE KZ'S ARE TO BE CALCULATED.
00917      C
00918      C      KZ IS USUALLY CALCULATED FROM THE EQUATION:
00919      C      KZ=VKAR*USTAR*ZH/PHI
00920      C      WHERE USTAR IS THE FRICTION VELOCITY AS CALCULATED BY THE
00921      C      EXACT SOLUTIONS TO:
00922      C      USTAR=VKAR*U/( INTEGRAL FROM Z0 TO ZW OF PHI/Z DZ)
00923      C      AND PHI IS THE PHI-FUNCTION FOR STABILITY CORRECTION:
00924      C      PHI=1+4.7*ZH/ZLEN FOR STABLE
00925      C      PHI=(1-15*ZH/ZLEN)**(-0.25) FOR UNSTABLE
00926      C      FOR UNSTABLE SURFACE CONDITIONS AND ZH GE -5*ZLEN, KZ IS :
00927      C      KZ=0.5*USTAR*ZH*(-0.4*ZH/(VKAR*ZLEN))**(1/3)
00928      C
00929      C      FOR NEUTRAL SURFACE CONDITIONS, ZLEN APPROACHES INFINITY, AND
00930      C      THE PHI-FUNCTION APPROACHES UNITY, YIELDING:
00931      C      KZ=VKAR*USTAR*ZH
00932      C
00933      C      IF THERE IS AN ELEVATED INVERSION LAYER, THE KZ'S CALCULATED
00934      C      FOR NEUTRAL OR UNSTABLE SURFACE CONDITIONS ARE "ROLLED OFF"
00935      C      TO A SMALL VALUE NEAR THE BASE OF THE INVERSION LAYER,
00936      C      ACCORDING TO THE EQUATION:
00937      C      KZ=KZ*(1.1-ZH/ZIBSE)
00938      C      FOR 0.1*ZIBSE LE ZH LE 1.1*ZIBSE.
00939      C      OBVIOUSLY THIS DOES NOT APPLY FOR STABLE SURFACE LAYERS.
00940      C
00941      C      IN THE INVERSION LAYER, ZLEN AND U ARE ASSUMED EQUAL TO
00942      C      20 AND 3, RESPECTIVELY, REGARDLESS OF THE VALUES INDICATED
00943      C      BY SURFACE MEASUREMENTS. ZLEN=20 AND U=3 ARE VALUES CONSISTENT
00944      C      WITH THOSE WHICH WOULD BE EXPECTED TO BE FOUND IN INVERSION
00945      C      LAYERS, AND WILL YIELD SMALL VERTICAL DIFFUSIVITIES.
00946      C
00947      C      ABOVE A STABLE LAYER, WHETHER IT IS AT THE SURFACE OR ALOFT,
00948      C      THE ATMOSPHERE IS ASSUMED TO BE NEUTRALLY STRATIFIED, WITH
00949      C      ZLEN = 1.E+05 AND U RESET TO THAT MEASURED AT THE SURFACE.
00950      C
00951      C      IN EITHER AN ELEVATED STABLE OR NEUTRAL LAYER, ZH IS USED
00952      C      UNALTERED, WHICH CAN BE INTERPRETED BY THE MODEL AS INDICATING

```

```

00953 C THAT THE STABILITY REGIME FOR WHICH IT IS CURRENTLY PERFORMING
00954 C CALCULATIONS, EXISTED FROM THE GROUND UP TO THAT POINT.
00955 C
00956 C ZLEN LT 0 IMPLIES THAT THE ATMOSPHERE IS UNSTABLE. ZLEN GT 0
00957 C IMPLIES THAT THE ATMOSPHERE IS STABLE. AS THE ABSOLUTE VALUE
00958 C OF ZLEN APPROACHES INFINITY, THE ATMOSPHERE IS CONSIDERED
00959 C NEUTRALLY STABLE.
00960 C
00961 C THE FOLLOWING ROUTINE IS NOT PRESENTLY PROGRAMMED TO HANDLE
00962 C MULTIPLE INVERSION LAYERS.
00963 C
00964 C
00965 C
00966 C Z LOOP
00967 DO 10285 K=1,MZ
00968 C
00969 C
00970 WSPD = SPEED(I,J)
00971 ZH1=ZC(K)
00972 IF(ZIBSE(I,J).LT.-1. ) GO TO 10240
00973 IF(ZC(K).LE.(1.1*ZIBSE(I,J))) GOTO 10240
00974 IF(ZC(K).GT.ZITOP(I,J)) GOTO 10230
00975 IF(ZIBSE(I,J).LT.1.) GOTO 10240
00976 ZLEN=20.
00977 WSPD=3.
00978 ZH1=ZC(K)-ZIBSE(I,J)
00979 GOTO 10240
00980 10230 ZLEN=-1.E+05
00981 ZH1=ZC(K)-ZITOP(I,J)
00982 10240 IF(ZLEN.LT.0.) GOTO 10250
00983 USTAR=VKAR*WSPD/( ALOG(ZW/Z0V)+4.7*(ZW-Z0V)/ZLEN)
00984 PHI=1.+4.7*ZC(K)/ZLEN
00985 DIFF=VKAR*USTAR*ZC(K)/PHI
00986 GOTO 10280
00987 10250 ZW4=(1.-15.*ZW/ZLEN)**0.25
00988 Z04=(1.-15.*Z0V/ZLEN)**0.25
00989 USTAR=VKAR*WSPD/( ALOG((ZW4-1.)/(Z04+1.))+2.*ATAN(ZW4)
00990 1 -ALOG((Z04-1.)/(Z04+1.))-2.*ATAN(Z04))
00991 IF(ZC(K).LT.(-5.*ZLEN)) GOTO 10260
00992 DIFF=0.5*USTAR*ZC(K)*(-0.4*ZC(K)/(VKAR*ZLEN))
00993 1 **0.333333
00994 GOTO 10270
00995 10260 PHI=(1.-15.*ZC(K)/ZLEN)**(-0.25)
00996 DIFF=VKAR*USTAR*ZC(K)/PHI
00997 10270 IF(ZIBSE(I,J).LE.0.) GOTO 10280
00998 IF(ZC(K).GE.(0.1*ZIBSE(I,J)).AND.ZC(K).LE.(1.1*ZIBSE(I,J)))
00999 1 DIFF=DIFF*(1.1-ZC(K)/ZIBSE(I,J))
01000 10280 IF(DIFF.GE.1000.) DIFF=1000.
01001 DMAX(K)=AMAX1(DMAX(K),DIFF)
01002 DMIN(K)=AMIN1(DMIN(K),DIFF)
01003 D(I,J,K)=DIFF
01004 10285 CONTINUE
01005 C END OF Z LOOP
01006 10290 CONTINUE
01007 C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
01008 WRITE(6,777)(DMAX(K),K=1,MZ),(DMIN(K),K=1,MZ)

```



```

01065      IF( NOTE.EQ.1) WRITE(6,2005) (K,K=1,MZ)
01066      IF( NOTE.EQ.1) WRITE(6,2006) (ZC(K),K=1,MZ)
01067      NOTE=0
01068      2005 FORMAT('1',T22,'WIND SPEED-UP PROFILES (M/S)'/1X,
01069      1T22,28(' ')//1X,T30,'K = ',20(12,4X))
01070      2006 FORMAT(1X,T14,'CELL CENTER HT(MD = ',20(F5.1,1X),T6,'SITE #')
01071      WRITE(6,2007) N, IN,JN,(WS(N,IO),K=1,MZ)
01072      2007 FORMAT(/1X,T9,I2,T15,'I = ',I2,' J = ',I2,' SPEED = ',
01073      1 20(F4.1,2X))
01074      210 CONTINUE
01075      C
01076      *****
01077      C----READ PIBAL DATA
01078      C----ASSIGN TO CELL CENTERS
01079      C----MERGE WITH GROUND WINDS
01080      *****
01081      C----COUNT PIBAL SITES
01082      C
01083      NP=0
01084      315 NP=NP+1
01085      3113 READ(5,722) ITYPE
01086      IF( ITYPE.EQ.42) GO TO 3113
01087      IF( ITYPE.EQ.81) GO TO 9021
01088      READ(5,191) ITYPE,III,JJJ,IDATE(3),IDATE(1),IDATE(2),IHR,NCARD,
01089      1ZP(1),Z42,WDV(1),WSV(1),ZP(2),WDV(2),WSV(2)
01090      191 FORMAT(I2,2I3,5I2,2(2X,2F4.0),4X,F4.0,6X,2F4.0)
01091      C
01092      C
01093      C----CHECK CARD TYPE
01094      C
01095      IF( ITYPE.NE.81.AND. ITYPE.NE.42) GO TO 9011
01096      IF( ITYPE.EQ.81) GO TO 9021
01097      C
01098      C----LOAD TIME
01099      C
01100      MDATE(1) = IDATE(1)
01101      MDATE(2) = IDATE(2)
01102      MDATE(3) = IDATE(3)
01103      MHR = IHR
01104      C
01105      C----CHECK DATE
01106      C
01107      IF( NDATE(1).EQ. MDATE(1).AND. NDATE(2).EQ. MDATE(2).AND. MDATE(3)
01108      1. EQ. NDATE(3).AND. NHR.EQ. MHR) GO TO 211
01109      WRITE(6,9010)
01110      WRITE(6,9010)
01111      STOP 24
01112      C----MESSAGE ABOUT WRONG TIME
01113      9010 FORMAT('1','OUTPUT FROM PREMET'//'*','TIME ALIGNMENT INCORRECT,
01114      1 THE TIME AND DATE OF THE DATA INPUT MUST AGREE WITH THE SPECI
01115      2-FICATION OF NDATE AND NHR IN THE DATA STATEMENT'//'*','THE
01116      3 TIME ON THE CARD WAS',4(I3,' ',''), 'THE TIME REQUESTED WAS NHR,
01117      4 NDATE= ',4(I3,' ',''), 'THIS OCCURRED FOR CARD NO.',I3)
01118      C----CONVERT TO GROUND LEVEL
01119      C
01120      211 CONTINUE

```

```

01121      ZG(1)=ZP(1)-Z42
01122      ZG(2)=ZP(2)-Z42
01123      KV=2
01124      IF(NCARD.LE.1) GO TO 301
01125      C
01126      C-----READ (NCARD-1) MORE CARDS
01127      C
01128      212 READ(5,191) ITYPE, IDUM, IDUM, IDUM, IDUM, IDUM, IDUM, IDUM,
01129      1Z1, DUM, WD1, WS1, Z2, WD2, WS2
01130      KV=2*(NCARD-NSEQ)+1
01131      ZP(KV)=Z1
01132      ZG(KV)=Z1-Z42
01133      WSV(KV)=WS1
01134      WDV(KV)=WD1
01135      KV=KV+1
01136      ZP(KV)=Z2
01137      ZG(KV)=Z2-Z42
01138      WSV(KV)=WS2
01139      WDV(KV)=WD2
01140      IF(ITYPE.NE.42) GO TO 9011
01141      IF(NSEQ.GT.1) GO TO 212
01142      C
01143      C-----CHECK 2ND DATUM
01144      C
01145      301 IF(WSV(KV).LT.-1.E-05) GO TO 302
01146      NOKV=2*NCARD
01147      GO TO 303
01148      302 NOKV=2*NCARD-1
01149      C
01150      C-----ECHO PIBAL DATA WITH RESPECT TO MSL
01151      C-----AS WELL AS GROUND
01152      C
01153      303 IF(NP.NE.1) GO TO 3066
01154      WRITE(6,305) MDATE(1),MDATE(2),MDATE(3),MHR,NDCOD(1,2),NDCOD(2,2)
01155      305 FORMAT('1','OUTPUT FROM PREMET'//I2,'/',I2,'/',I2,5X,'PIBAL DA
01156      1TA ECHO',5X,'HR=',I4,'PST',20X,'VERSION=',214)
01157      3066 WRITE(6,306)
01158      306 FORMAT(//2X,'PIBAL',7X,'I',4X,'J',7X,'SITE',3X,'ELEVATION',6X,
01159      1'ELEVATION',6X,'SPEED',6X,'DIR'/1X,'SITE',1X,'NO.',',
01160      12X,'(OUTER GRID)',4X,'ELEVATION'
01161      2,2X,'OF MEASUREMENT',2X,'OF MEASUREMENT'/40X,'M (MSL)',10X,'M (GRD
01162      3)',6X,'(M/S)',4X,'(DEGREES)',T28,'(METERS')/')
01163      DO 307 K=1,NOKV
01164      KN=NOKV-K+1
01165      WRITE(6,309) NP,III,JJJ,Z42,ZP(KN),ZG(KN),WSV(KN),WDV(KN)
01166      309 FORMAT(/3X,I2,8X,I2,4X,I2,9X,F4.0,10X,F5.0,8X,F5.0,7X,F4.0,
01167      17X,F4.0)
01168      307 CONTINUE
01169      C
01170      C-----SHIFT TO INNER GRID
01171      C
01172      IIX(NP)=III-MIX
01173      JJJ(NP)=JJJ-MIY
01174      C
01175      C-----ASSIGN DATA TO CELL CENTERS USING R**6 VERTICAL INTERPOLATION
01176      C

```

```

01177 C-----LOOP OVER CELL CENTERS
01178 DO 313 K=2,MZ
01179 SNOR=0.
01180 SUMDX=0.
01181 SUMDY=0.
01182 SUMSP=0.
01183 C
01184 C-----LOOP OVER VERTICAL DATA POINTS
01185 DO 312 KV=1,NOKV
01186 RSIX=ZC(K)-ZG(K)+1.E-10
01187 RTWO=RSIX*RSIX
01188 RSIX=RTWO*RTWO
01189 RSIX=RSIX*RTWO
01190 C
01191 C-----CALC. SPEED & DIRECTION SUMS
01192 C
01193 METDIR=WDV(KV)*3.1415926/180.
01194 SUMSP=WSV(KV)/RSIX+SUMSP
01195 SUMDX=-SIN(METDIR)/RSIX+SUMDX
01196 SUMDY=-COS(METDIR)/RSIX+SUMDY
01197 SNOR=1./RSIX+SNOR
01198 312 CONTINUE
01199 C
01200 C-----CALC. SPEED AND DIR AT CELL CENTER
01201 C
01202 PWS(NP,K)=SUMSP/SNOR
01203 WX=SUMDX/SNOR
01204 WY=SUMDY/SNOR
01205 METDIR=(ATAN2(WY,WX)*180./3.1415926
01206 PWD(NP,K)=AMOD((270.-METDIR),360.)
01207 313 CONTINUE
01208 C
01209 C-----PRINT PIBAL INTERPOLATED DATA
01210 C
01211 WRITE(6,3111)
01212 3111 FORMAT(//2X,'PIBAL',7X,'I',4X,'J',7X,'SITE',13X,'CELL CENTER',6X
01213 1,'CELL CENTER',9X,'SPEED',8X,'DIR'/1X,'SITE',1X,'NO.',
01214 12X,'(INNER GRID)',4X,'ELEVATION',' LEVEL',
01215 22X,'ELEVATION',7X,'ELEVATION'/47X,'M (MSL)',
01216 310X,'M (WRTG)',9X,'(N/S)',5X,'(DEGREES)',T28,'(METERS) /')
01217 DO 311 KK=2,MZ
01218 K=MZ-KK+2
01219 ZCG=ZC(K)+Z42
01220 WRITE(6,3091) NP,IIX(NP),JJY(NP),Z42,K,ZCG,ZC(K),
01221 1,PWS(NP,K),PWD(NP,K)
01222 3091 FORMAT('0'//3X,12,8X,I2,4X,I2,9X,F4.0,5X,I2,6X,F5.0,15X,F4.0, 9X,
01223 1F4.0,11X,F4.0)
01224 WRITE(6,3092) K,ZCG,ZC(K),PWS(NP,K),PWD(NP,K)
01225 3092 FORMAT(/39X,I2, 6X,F5.0,14X,F5.0, 9X,F4.0,11X,F4.0)
01226 311 CONTINUE
01227 WRITE(6,3093)
01228 3093 FORMAT(///5X,100('*'))
01229 GO TO 315
01230 9011 WRITE(6,9012) ITYPE
01231 C-----MESSAGE ABOUT WRONG TYPE
01232 9012 FORMAT(' PREMET::PROGRAM STOPPED BECAUSE A PIBAL SEQUENCE CARD HAD'

```

```

01233      1 THE WRONG TYPE, ITS TYPE WAS TYPE = ',G10.2)
01234      STOP 42
01235      NOPIBL=NP-1
01236      IF( NOPIBL.EQ.0) WRITE(6,9020)
01237      9020 FORMAT('1',T40,'NO PIBAL DATA AVAILABLE FOR THIS HOUR ... PREMET')
01238      C
01239      C-----R**2 INTERPOLATION OF GROUND AND PIBAL DATA TO YIELD FINAL U & V
01240      C
01241      C-----LOOP OVER VERTICAL CELLS AND GROUND CELLS
01242      DO 411   K=2,MZ
01243      DO 411   J=1,MY
01244      DO 411   I=1,MX
01245      ANORM=0.
01246      WXSUM=0.
01247      WYSUM=0.
01248      C
01249      C-----LOOP OVER GROUND DATA SITES
01250      C
01251      DO 412   NG=1,NOSITE
01252      IF(WS(NG,1).LT.-1.E-05) GO TO      412
01253      C
01254      C-----COMPUTE RSQ
01255      ANGLE=WD(NG)*3.1415926/180.
01256      XS= I-IX(NG)
01257      XS= J-JY(NG)
01258      RSQ=XS*XS+YS*YS+1.E-15
01259      ANORM=ANORM+1./RSQ
01260      WXSUM=WS(NG,K)*SIN(ANGLE)/RSQ+WXSUM
01261      WYSUM=WS(NG,K)*COS(ANGLE)/RSQ+WYSUM
01262      412 CONTINUE
01263      C
01264      C-----LOOP OVER PIBAL DATA
01265      IF( NOPIBL.LT.1) GO TO 7413
01266      DO 413   NP=1,NOPIBL
01267      C-----COMPUTE RSQ
01268      XS= I-IIX(NP)
01269      XS= J-JJY(NP)
01270      RSQ=XS*XS+YS*YS+1.E-15
01271      C
01272      C-----CALC. CONTRIBUTION OF PIBAL DATA
01273      ANGLE=PWD(NP,K)*3.1415926/180.
01274      ANORM=ANORM+1./RSQ
01275      WXSUM=WXSUM-PWS(NP,K)*SIN(ANGLE)/RSQ
01276      WYSUM=WYSUM-PWS(NP,K)*COS(ANGLE)/RSQ
01277      413 CONTINUE
01278      7413 CONTINUE
01279      U(I,J,K)=WXSUM/ANORM
01280      V(I,J,K)=WYSUM/ANORM
01281      411 CONTINUE
01282      C
01283      C-----PRINT THE WIND COMPONENTS
01284      C-----FIRST U THEN V
01285      WRITE(6,9025) NDCOD(1,2),NDCOD(2,2),MDATE,MR
01286      9025 FORMAT('1','OUTPUT OF CELL-CENTERED,X-COMPONENT,U(I,J,K',
01287      1'5X,'VERSION='',I2,'/,I2,10X,I2,'/,I2,'/,I2,5X,
01288      2'HR='',I2.'00PST')

```

```
01289      CALL OUT3D(U, MX, MY, MZ, 1)
01290      WRITE(6, 9030) NDCOD(1,2), NDCOD(2,2), MDATE, MHR
01291 9030 FORMAT('1', 'OUTPUT OF CELL-CENTERED, Y-COMPONENT, V(I,J,K)', 
01292      1 5X, 'VERSION=' , I2, '/', I2, 10X, I2, '/', I2, '/', I2.5X,
01293      2 'HR=' , I2, '00PST')
01294      CALL OUT3D(V, MX, MY, MZ, 2)
01295 C
01296 C-----WRITE THE WIND COMPONENTS TO TAPE 11
01297 C
01298      WRITE(11) (((U(I,J,K) , I=1, MX) , J=1, MY) , K=1, MZ) ,
01299      1(((V(I,J,K) , I=1, MX) , J=1, MY) , K=1, MZ)
01300      RETURN
01301 C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
01302 C      STOP 9999
01303 C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
01304      END
```

```

00001      SUBROUTINE OUT3D(ARRAY,NX,NY,NZ,NAME)
00002      DIMENSION ARRAY(NX,NY,NZ)
00003      C
00004      C    10TH TITLE IS A UTILITY BLANK FIELD
00005      C
00006      REAL*8 TITLE(10)
00007      DATA TITLE /'U          ','V          ','W(FACES)', 'EX-FACES',
00008      .'EY-FACES', 'EZ-FACES', 'SPEED', 'XY ANGLE', 'Z ANGLE', '/',
00009      C    IF(NAME.LT.4) WRITE( 6,201)
00010      C    IF(NAME.LT.7 .AND. NAME.GT.3) WRITE( 6,202)
00011      C201 FORMAT('1',T55,'OUTPUT FROM WIND''0',T55,16('*')/)
00012      C202 FORMAT('1',T55,'DIFFUSIVITIES''0',T55,13('*')/)
00013      N = (NX + 10)/11
00014      M = MOD(NX,11)
00015      DO 10 KK = 1,NZ
00016      K = NZ - KK + 1
00017      C    IF(KK.LT.7) KK = KK + 2
00018      203  WRITE( 6,203) K,TITLE(NAME)
00019      FORMAT(//'*OK= ',I2,T60,A8,'CELL-CENTERED...(M/S)')
00020      DO 10 ICOOUNT = 1,N
00021      I2 = ICOOUNT*11
00022      I1 = I2 - 10
00023      IF(ICOOUNT.EQ.N .AND. M.GT.0) I2 = I1 + M - 1
00024      WRITE( 6,205)(II,II = I1,I2)
00025      DO 10 J = 1,NY
00026      JJ = NY - J + 1
00027      10   WRITE( 6,204) JJ,(ARRAY(I,JJ,K),I = I1,I2)
00028      204  FORMAT('0 J=',I2,3X,11(1PE11.2))
00029      205  FORMAT('/'0',8X,11(16,5X))
00030      RETURN
00031      END

```

```
00001      FUNCTION JULIAN(IYR, IMO, IDY)
00002      C
00003      C      THE JULIAN FUNCTION CALCULATES THE JULIAN DATE GIVEN THE YEAR
00004      C      (LAST TWO DIGITS ONLY), THE MONTH, AND THE DAY. IT CHECKS FOR
00005      C      A LEAP YEAR AND CHANGES THE NUMBER OF DAYS IN FEBRUARY
00006      C      ACCORDINGLY.
00007      C
00008      DIMENSION IDAY(12)
00009      DATA IDAY/31,28,31,30,31,30,31,31,30,31,30,31/
00010      JULN=0
00011      LEAP=0
00012      IF(MOD(IYR,4).EQ.0) LEAP=1
00013      M=IMO-1
00014      IF(M.LE.0) GOTO101
00015      DO 100 I=1,M
00016      JULN=JULN+IDAY(I)
00017      IF(LEAP.EQ.1.AND.I.EQ.2) JULN=JULN+1
00018      100 CONTINUE
00019      101 JULIAN=JULN+IDY
00020      RETURN
00021      END
```

```

00001      SUBROUTINE OUT3D2(ARRAY,NX,NY,NZ,NAME)
00002      INTEGER ARRAY
00003      DIMENSION ARRAY(NX,NY,NZ)
00004      REAL*8 TITLE(10)
00005      DATA TITLE /'U          ','V          ','W(FACES) ','EX-FACES',
00006      .'EY-FACES','EZ-FACES','SPEED','XY ANGLE','Z ANGLE','
00007      C      IF(NAME.LT.4) WRITE( 2,201)
00008      C      IF(NAME.LT.7 .AND. NAME.GT.3) WRITE( 2,202)
00009      C201    FORMAT('1',T55,'OUTPUT FROM WIND',//0',T55,16('*')/)
00010      C202    FORMAT('1',T55,'DIFFUSIVITIES',//0',T55,13('*')/)
00011      N = (NX + 10)/11
00012      M = MOD(NX,11)
00013      DO 10  KK = 1,NZ
00014      K = NZ - KK + 1
00015      C      IF(KK.LT.7) KK = KK + 2
00016      WRITE( 6,203) K,TITLE(NAME)
00017      203    FORMAT(///'OK= ',I2,T60,A8,'CELL-CENTERED...(M/S)')
00018      DO 10 ICOUNT = 1,N
00019      I2 = ICOUNT*11
00020      I1 = I2 - 10
00021      IF(ICOUNT.EQ.N .AND. M.GT.0) I2 = I1 + M -1
00022      WRITE( 6,205)(II,II = I1,I2)
00023      DO 10 J = 1,NY
00024      JJ = NY - J + 1
00025      10     WRITE( 6,204) JJ,(ARRAY(I,JJ,K),I = I1,I2)
00026      204    FORMAT( '0 J= ',I2,8X,11(I11))
00027      205    FORMAT(//0',8X,11(I6,5X))
00028      RETURN
00029      END

```

```

00001      SUBROUTINE COAST(F,NOSITE,MX,MY,II,JJ,FMAX,FMIN,ALP,F2D)
00002          DIMENSION ICOAST(60),F(1),F2D(MX,MY),II(1),JJ(1),IBC1(60)
00003          COMMON/TERRA/IBC1,ICOAST
00004          COMMON/GRID0/NX,NY,NZ,MIX,MIY
00005      C
00006      C -----LOOP OVER CELLS
00007      C
00008          FMAX=-400.
00009          FMIN=100000.
00010          NODAT = 0
00011          DO 1 J=1,MY
00012          DO 1 I=1,MX
00013          ANORM=0.
00014          FSUM=0.
00015      1111 DO 2 N=1,NOSITE
00016          IF(F(N).LT.-1.E-05) GO TO 22
00017          FMAX=AMAX1(FMAX,F(N))
00018          JP=J+MIY
00019          JS=JJ(N)+MIY
00020          FMIN=AMIN1(FMIN,F(N))
00021          WTSQD=ALP*(I- ICOAST(JP)-II(N)+ICOAST(JS ))**2+1.E-15+
00022          1(J - JJ(N))**2
00023          ANORM=ANORM+1./WTSQD
00024          FSUM=FSUM+F(N)/WTSQD
00025          GO TO 2
00026      22 NODAT = NODAT + 1
00027      2 CONTINUE
00028          IF(NODAT.EQ.NOSITE) RETURN
00029          F2D(I,J)=FSUM/ANORM
00030      1 CONTINUE
00031          DO 10 J=1,MY
00032          JP=J+MIY
00033          IC= ICOAST(JP)-MIX
00034          IT= IC-1
00035          IF(IT.LE.0) GO TO 10
00036          SET=F2D(IC,J)
00037          DO 10 I=1,IT
00038          F2D(I,J)=SET
00039      10 CONTINUE
00040          RETURN
00041          END

```

```

00001      SUBROUTINE OUTPUT(ARRAY, MX, MY, MIX, MIY, XMAX, IEC)
00002      C
00003      C ..... OUTPUT IS CALLED TO PRINT ANY MX*MY ARRAY
00004      C ..... USING AN OFFSET OF MIX AND MIY AND GIVEN THE
00005      C ..... MAXIMUM VALUE IN THE ARRAY = XMAX (WHICH MUST BE POSITIVE)
00006      C
00007      C
00008      C ..... PRINT THE EULERIAN CONCENTRATIONS
00009      C           DIMENSION ICE( 50),           ARRAY(MX, MY), IBC(MY)
00010      C
00011      C ..... FIND FACTOR FOR A TWO DIGIT PRINT .....
00012      C
00013      F = 1.
00014      FACTOR = 1.
00015      IF(XMAX.LT.1.E-30) GO TO 11
00016      LOGX = IFIX(ALOG10(XMAX))
00017      IF(XMAX .LT. 1.0) LOGX = LOGX - 1
00018      FACTOR = 10.**(LOGX - 1)
00019      F = 1./FACTOR
00020      11 CONTINUE
00021      102 FORMAT('0',T47,'MULTIPLY VALUES IN THIS TABLE BY',1PE8.1)
00022      WRITE(6,102) FACTOR
00023      MOS=MIX+1
00024      MOT=MX+MIX
00025      103 FORMAT(' ',5X,42(I3))
00026      WRITE(6,103) (I,I=MOS,MOT)
00027      WRITE(6,103) (I,I = 1, MX)
00028      C
00029      C ..... PRINT CONCENTRATIONS .....
00030      C
00031      DO 30 JJ = 1, MY
00032      J = MY - JJ + 1
00033      JL=J+MIY
00034      LX = IBC(J)
00035      DO 15 I = 1, LX
00036      A = ARRAY(I,J)*F + 0.5
00037      IF(A.LT.0.5) A = A - 1.0
00038      15 ICE(I) = IFIX(A)
00039      104 FORMAT('0',I2,1X,I2,42(I3))
00040      WRITE(6,104) JL,J,(ICE(I),I = 1, LX)
00041      30 CONTINUE
00042      RETURN
00043      END

```

```

00001 C
00002 C SUBROUTINE PL(Z,NX,NY,ZMAX,ZMIN)
00003 C*****
00004 C*
00005 C* P L IS A PRINTER-PLOTTER ROUTINE WHICH *
00006 C* SCALES MATRIX VALUES FROM 0 TO 9 *
00007 C*
00008 C*****
00009 C
00010 DIMENSION LQ(72),L(131),Z(NX ,NY)
00011 DATA NNC,AMAX,AMIN,LOGFL/10,0.,0.,0/
00012 DATA LQ / '0', '1', '2', '3', '4', '5',
00013 . , '6', '7', '8', '9', 'A', 'B',
00014 . , 'C', 'D', 'E', 'F', 'G', 'H',
00015 . , 'J', 'K', 'L', 'M', 'N', 'O',
00016 . , 'P', 'Q', 'R', 'S', 'T', 'U',
00017 . , 'W', 'X', 'Y', 'Z', '/',
00018 DATA ISTAR/*/,IMAX/60/
00019 C
00020 C
00021 C
00022 LOGFL = 0
00023 AMIN = 0.0
00024 AMAX = 0.0
00025 NNC = 10
00026 NXMX=NX
00027 C ASPECT = DY/DX
00028 ASPECT = 1.0
00029 JMAX = .6*NY/NX*IMAX*ASPECT +.5
00030 NC = MIN0(NNC,36)
00031 ZMIN = AMIN
00032 ZMAX = AMAX
00033 ANX = NX
00034 ANY = NY
00035 IF (ABS(ZMAX-ZMIN) .GT. 1.0E-5) GO TO 40
00036 ZMAX = Z(1,1)
00037 ZMIN = ZMAX
00038 DO 20 I=1,NX
00039 DO 20 J=1,NY
00040 ZMAX = AMAX1(ZMAX,Z(I,J))
00041 20 ZMIN = AMIN1(ZMIN,Z(I,J))
00042 IF (ABS(ZMAX-ZMIN) .LT. 1.0E-20) RETURN
00043 40 IF (LOGFL .EQ. 0) GO TO 80
00044 IF (ZMIN .GT. 0.) GO TO 60
00045 WRITE (6,220)
00046 RETURN
00047 60 HZMAX = ZMAX
00048 HZMIN = ZMIN
00049 ZMAX = ALOG10(ZMAX)
00050 ZMIN = ALOG10(ZMIN)
00051 80 SC = (NC*2.-1.E-5)/(ZMAX-ZMIN)
00052 DX = ANX/IMAX
00053 DY = ANY/JMAX
00054 Y = ANY+DY
00055 DO 180 JJ=1,JMAX
00056 J = JMAX -JJ +1

```

```

00057      Y = Y-DY
00058      M = Y + .5
00059      DM = AMAX1(Y - M + .5, 1.0E-10)
00060      IF(M.EQ.0) DM = 1.0
00061      IF(M.GE.NY) DM = 0.0
00062      X = 0.
00063      DO 160 I=1, IMAX
00064      X = X+DX
00065      IF( (I*J .EQ. 1 .OR. I*j .EQ. IMAX*jMAX) GO TO 140
00066      IF( (I .EQ. 1 .AND. J .EQ. JMAX .OR. J .EQ. 1 .AND. I .EQ. IMAX)
00067      GO TO 140
00068      N = X + .5
00069      DN = AMAX1(X - N + .5, 1.0E-10)
00070      IF(N.LE.0) DN = 1.0
00071      IF(N.GE.NX) DN = 0.0
00072      IF( LOGFL .NE. 0) GO TO 100
00073      DMM = AMAX1(1. - DM, 1.0E-10)
00074      DNN = AMAX1(1. - DN, 1.0E-10)
00075      IF(M.EQ.0) DMM = 0.0
00076      IF(M.LE.0) M=1
00077      IF(N.EQ.0) DNN = 0.0
00078      IF(N.LE.0) N=1
00079      NP1=N+1
00080      IF(NP1.GT.NX) NP1=NX
00081      MP1=M+1
00082      IF(MP1.GT.NY) MP1=NY
00083      C = DN*(DM*Z(NP1,MP1)+DMM*Z(NP1,M))+DNN*(DM*Z(N,MP1)+DMM*
00084      .Z(N,M))
00085      GO TO 120
00086      100 C = DN*(DM* ALOG10(Z(N+1,M+1))+(1.-DM)* ALOG10(Z(N+1,M))+ (1.-DN)*
00087      .(DM* ALOG10(Z(N,M+1))+(1.-DM)* ALOG10(Z(N,M)))
00088      120 IND = 1+SC*(C-ZMIN)
00089      IF( (IND .LE. 0) IND = 1
00090      IF( (IND .GT. 2*NC) IND = 2*NC
00091      L(I) = LQ(IND)
00092      GO TO 160
00093      140 L(I) = ISTAR
00094      160 CONTINUE
00095      C      INDEX CHANGE TO MAKE PRINTOUT IN MATRIX - LIKE FORM
00096
00097
00098      180 WRITE (6,240) (L(I), I = 1 , IMAX)
00099      IF (LOGFL .EQ. 0) GO TO 200
00100      ZMAX = HZMAX
00101      ZMIN = HZMIN
00102      200 WRITE (6,260) ZMIN,ZMAX
00103      C      RETURN
00104      220 FORMAT (/////
00105      115H ****NON-POSITIVE VALUE IN ARRAY TO BE
00106      . CONTOURED LOGARITHMICALLY. PL IS RETURNING WITHOUT PLOTTING )
00107
00108      C      240 FORMAT (1X,13A1)
00109      260 FORMAT (/17H0MINIMUM VALUE = ,1PE9.3,5X,16HMAXIMUM VALUE = ,E9.3)
00110
00111      280 FORMAT(1H1)
00112      END

```

### A.3.3 PRESOR

```
00001 C THIS DRIVES PRESOR UNDER THE SAME INPUT FILE STRUCTURE
00002 C ASSUMPTIONS AS WERE USED FOR PREBIC
00003 C
00004 C
00005 C DIMENSION NDATE(3), IBC1(60), IC1(60)
00006 C INTEGER DAY, YEAR, BEGIN
00007 C COMMON/TERRA/ IBC1, IC1
00008 C COMMON/GRID0/MX, MY, MZ, MIX, MIY
00009 C NAMELIST/TERR0/ IBC1, IC1
00010 C NAMELIST/GRID0/MX, MY, MZ, MIX, MIY
00011 C NAMELIST /SETUP/ MONTH, DAY, YEAR, BEGIN, NOHRS
00012 C
00013 C
00014 READ(5,SETUP)
00015 WRITE(6,SETUP)
00016 READ(5,GRID0)
00017 WRITE(6,GRID0)
00018 READ(5,TERR0)
00019 WRITE(6,TERR0)
00020 NDATE(1) = MONTH
00021 NDATE(2) = DAY
00022 NDATE(3) = YEAR
00023 IHR = BEGIN
00024 DO 10 N = 1, NOHRS
00025 1 READ(5,101) ITYPE
00026 IF( ITYPE.NE.81) GO TO 1
00027 CALL PRESOR(NDATE, IHR)
00028 IHR = IHR + 1
00029 IF( IHR.LT.24) GO TO 10
00030 IHR = 0
00031 NDATE(2) = NDATE(2) + 1
00032 10 CONTINUE
00033 STOP
00034 101 FORMAT( I2)
00035 END
```

```

00001      SUBROUTINE PRESOR(NDATE,NHR)
00002      C PRESOR READS EMISSION RATES (KG/HR) AND OUTPUTS TOTAL EMISSIONS
00003      C FOR EACH CELL
00004      C
00005      C THE INPUT IS CO, NOX AND THC
00006      C THE OUTPUT IS CO, NO2, NO, HC1, HC2, HC3
00007      C
00008      C VARIABLES OF SOME INTEREST ARE: MX, MY, MZ ARE THE INNER GRID
00009      C DIMENSIONS STORE (MX, MY, MS) - ACCUMULATES THE GRD EMISSIONS
00010      C FROM FOLLOWING TYPES (I.E. MV + RAIL + GROUND LEVEL POINT SOURCES
00011      C + AREA + SHIPPING)
00012      C
00013      C S(MX, MY, MS) - WORKING ARRAY WHICH ACCUMULATES EMISSIONS FOR
00014      C ONE SOURCE TYPE AT A TIME. S IS PRINTED AND REZEROED AFTER
00015      C PROCESSING EACH SOURCE TYPE. (PPM/MIN)
00016      C
00017      C SE(MS, MX, MY, MZ) - SE IS THE FINAL ARRAY HOLDING THE SUM OF
00018      C ALL EMISSIONS BY SPECIE AND BY CELL. SE IS THE EMISSION ARRAY
00019      C USED BY MADCAP (PPM/MIN)
00020      C
00021      C C(3) - HOLDS CO, NOX AND THC AS READ OFF EACH CARD IMAGE (KG/HR)
00022      C
00023      C S1(6) - HOLDS CO, NO2, NO, HC1, HC2, HC3 AFTER PROCESSING OF
00024      C C(3) (PPM/MIN)
00025      C
00026      C F(13) - HOLDS THE THC FRACTIONS AS READ OFF EACH CARD IMAGE
00027      C FOR AREA AND GROUND LEVEL POINT SOURCES
00028      C
00029      C SPLIT(3, 17) - CONTAINS PERMANENT DATA DELINEATING THE THC
00030      C SPLIT INTO HC1, HC2, AND HC3 FOR EACH OF 17 TRW SOURCE TYPES
00031      C * * *
00032      *****DIMENSION S1(6),STORE(37,51, 6), SE(6,37,51,5), IC1(60)
00033      1 ,S(37,51, 6), C(3), F(13), DZ(5), NFT(13), HT(5),
00034      2 SPLIT(3,18), NDATE(3), MDATE(3), NAME(6), NDGCD(2,3),
00035      3 ,SMAX(6), SMIN(6), IBC(51), IBC1(60), WIND (37,51), KSTAB(37,51)
00036      COMMON/TERRA/IBC1, IC1
00037      COMMON/GRIDA/MX, MY, MZ, MIX, MIY
00038      DATA SPLIT / .0068, .0140, .000,
00039      1       .0049, .0180, .000,
00040      2       .0024, .0043, .0029,
00041      3       .0080, .0073, .0024,
00042      4       .0004, .0180, .000,
00043      5       .0005, .0210, .0004,
00044      5       .0005, .0210, .0004,
00045      6       .0000, .0300, .0000,
00046      7       .0000, .0000, .0000,
00047      8       .0000, .0200, .0000,
00048      9       .0000, .0300, .0000,
00049      1       .0008, .0170, .0008,
00050      6       .0000, .0300, .0000,
00051      3       .0095, .0170, .0000,
00052      4       .0043, .0180, .0000,
00053      5       .0032, .0160, .0056,
00054      2       .0062, .0150, .0000,
00055      1       .0065, .0150, .0031/
00056

```

```

00057      DATA NAME  /' CO ', ' NO2 ', ' NO ', ' HC1 ', ' HC2 ', ' HC3 '/
00058      DATA          SMAX, SMIN/6*-1., 6*1000./
00059  *****SITE SPECIFIC DATA*****
00060  *****DATA HT /50., 125., 250., 500., 1000./
00061  DATA DX, DY/2*2000./, IFLGC/2/
00062  *****DATA DX, DY/2*2000./, IFLGC/2/
00063  ****
00064      ICNT = 0
00065      NOTE1 = 0
00066      SUB=0.
00067      DO 1999 I=1,MZ
00068      TOP=HT(I)
00069      DZ(I)=TOP-SUB
00070  1999 SUB=TOP
00071      VOL1=DX*DY*DZ(1)
00072      CF=293.*1.E06*22.4/(60.*273.*VOL1)
00073  ****
00074  ****
00075      READ(14) ((WIND(I,J), I=1,MX), J=1, MY), ((KSTAB(I,J), I=1,MX), J=1, MY)
00076  ****
00077  ****
00078  ****
00079  ****      O R , S E T   S P E E D & S T A B
00080      SPEED = 3.
00081      STAB = 5.
00082  ****
00083  ****
00084  ****
00085      C . . . SHIFT MOUNTAIN BOUNDARIES TO THE INNER GRID . . . . . 853 C
00086      DO 27 NN = 1, MY
00087      27 IBC(NN) = MIN0(IBC1(NN+MIY) - MIX, MX)
00088  C
00089  C-----ZEROS S, STORE . SE
00090      DO 5 I=1,MX
00091      DO 5 J=1, MY
00092      DO 5 L=1,6
00093      S(I,J,L)=0.
00094      STORE(I,J,L)=0.
00095      DO 5 K = 1, MZ
00096      5 SE(L, I, J, K) = 0.0
00097  C
00098  C-----READ MOTOR VEHICLE DATA
00099      1 READ(5, 101) ITYPE, II, JJ, MDATE(3), MDATE(1), MDATE(2), MHR, (C(L), L=1,
00100      1 3), F1, F2, F3
00101      101 FORMAT( I2, 2I3, 4I2, 3F6.0, 13F3.3)
00102  C-----CHECK CARD TYPE
00103      IF( ITYPE.NE.81.AND. ITYPE.NE.82) GO TO 900
00104      IF( ITYPE.EQ.82) GO TO 3
00105      ICNT = 1
00106  C-----CHECK DATE
00107      IF( MHR.EQ. NHR. AND. MDATE(1).EQ. NDATE(1).AND. MDATE(2).EQ. NDATE(2).
00108      1 AND. MDATE(3).EQ. NDATE(3)) GO TO 2
00109  C-----MESSAGE ABOUT WRONG TIME
00110      WRITE(6, 102) MHR, MDATE, NHR, NDATE
00111      102 FORMAT('1', 'OUTPUT FROM PRESOR'// '0', 'TIME ALIGNMENT INCORRECT,
00112      1 THE TIME AND THE DATE OF DATA INPUT MUST AGREE WITH THE SPECIFI-

```

```

00113      2CATION OF NDATE AND NHR IN THE DATA STATEMENT'//', 'THE TIME ON
00114      3THE CARD WAS', 4(I3, ','), 'THE TIME REQUESTED WAS NHR, NDATE= ', 4(I3,
00115      4, ','), 'THIS OCCURED FOR CARD NO.', I3)
00116      STOP 24
00117 C-----ECHO DATA
00118      2 IF(NOTE1.EQ.0) WRITE(6,103) MDATE(1),MDATE(2),MDATE(3),MHR
00119      103 FORMAT('1',40X,'MOTOR VEHICLE EMISSIONS',5X,I2,'/',I2,'/',I2,5X,
00120      1'HR=',I4,'00 PST'
00121      1      '/40X,55('*')//10X,'CARD ECHO    KC/HR',T65,'ECHO AF
00122      2TER SPLITS IN (PPM/MIN) TIMES 1000'/10X,19('=',),T65,42('=')
00123      5      '/' I J CO NOX THC F14 F15 F1
00124      36',T62,'I J ',T69,'CO',5X,'NO2',5X,'NO',6X,'HC1',5X,'HC2',5X,'HC
00125      43'
00126      5      '/' == == === == == == == == == ==
00127      3= ',T62,'= = ',T69,'==',5X,'==',5X,'==',6X,'==',5X,'==',5X,'==
00128      4= ')
00129      NOTE1=1
00130      104 FORMAT(1X,I2,I2,1X,I2,F6.0,F6.1,F6.1,F6.2,F6.2,F6.2)
00131 C
00132 C - - - - SUPPRESS CARD ECHO ; COMMENT THE FOLLOWING WRITE
00133 C
00134 C      WRITE(6,104) II,JJ, (C(L),L=1, 3),F1,F2,F3
00135 C-----CONVERT TO PPM/MIN
00136 C-----SURFACE CELL VOLUME IS DX*DY*DZ(1)
00137 C-----SPLIT AND REARRANGE NOX AND HYDROCARBONS,X IS THE WT RATIO NO2/NOX
00138 X=0.01
00139 S1(2)=C(2)*2.0*X/(3.0-X)*CF/46.
00140 S1(3)=C(2)*2.0*(1.0-X)/(3.0-X)*CF/30.
00141 C-----HYDROCARBONS SPLIT
00142 S1(4)=C(3)*(F1*SPLIT(1,14)+F2*SPLIT(1,15)+F3*SPLIT(1,16))*CF
00143 S1(5)=C(3)*(F1*SPLIT(2,14)+F2*SPLIT(2,15)+F3*SPLIT(2,16))*CF
00144 S1(6)=C(3)*(F1*SPLIT(3,14)+F2*SPLIT(3,15)+F3*SPLIT(3,16))*CF
00145 S1(1)=C(1)*CF/28.
00146 C-----ECHO DATA IN PPM/MIN
00147 C-----PPM ECHO
00148 C
00149 C - - - - SUPPRESS ECHO BY COMMENTING THE FOLLOWING WRITE
00150 C
00151 C      WRITE(6,105) II,JJ, (S1(L),L=1,6)
00152 105 FORMAT(' ',T61,I2,1X,I2, 3PF6.1,5(3PF6.3))
00153 1051 FORMAT('0',I2,1X,I2,1X,11,65X,(F5.1,1X))
00154 C----- LOAD EMISSIONS INTO WORKING ARRAY S(I,J,L) AND SHIFT TO INNER GRID
00155 INEW=II - MIX
00156 JNEW=JJ - MIY
00157 IF(INEW.GT.0.AND.JNEW.GT.0.AND.INEW.LE.MX.AND.JNEW.LE.MY) GO TO 39
00158 406 FORMAT('/1X,' SOURCE ABOVE IS OUTSIDE THE INNER GRID AT IJ= ',2I3')
00159 C
00160 C - - - - SUPPRESS PRINTS OF EXTERNAL SOURCES
00161 C
00162 C      WRITE(6,406) II,JJ
00163 C      GO TO 1
00164 39 CONTINUE
00165 C-----LOAD INTO S( I, J,L)
00166 DO 40 L=1,6
00167 IF(S1(L).GT.0.0) S(INEW,JNEW,L)=S1(L)+S(INEW,JNEW,L)
00168 SMAX(L)=AMAX1(SMAX(L),S(INEW,JNEW,L))

```

```

00169 40 SMIN(L)=AMIN1(SMIN(L),S(INEW,JNEW,L))
00170 GO TO 1
00171 3 NOTE1 = 1
00172 IF( ICNT.EQ.0) GO TO 51
00173 DO 41 L=1,6
00174 107 FORMAT('1OUTPUT FROM PRESOR',T70,'DATE=',I2,'/',I2,'/',I2,' HR= ',I
00175 12,'00 PST')
00176 1107 FORMAT('1X,T45,'MOTOR VEHICLE EMISSIONS ',4X,A4,'(PPM/MIN)')
00177 WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),NHR
00178 WRITE(6,1107) NAME(L)
00179 CALL OUTPUT(S(1,1,L),MX,MY,MIX,NIY,SMAX(L),IBC)
00180 WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),NHR
00181 WRITE(6,1107) NAME(L)
00182 41 CALL PL(S(1,1,L),MX,MY,SMAX(L),SMIN(L))
00183 C-----STORE AND ZERO S
00184 DO 45 I=1,MX
00185 DO 45 J=1,MY
00186 DO 45 L=1,6
00187 STORE(I,J,L)=STORE(I,J,L)+S(I,J,L)
00188 45 S(I,J,L)=0.
00189 DO 46 L = 1,6
00190 SMIN(L) = 10000.
00191 46 SMAX(L) = -1.
00192 ICNT = 0
00193 GO TO 51
00194 900 WRITE(6,9001) ITYPE
00195 STOP 71
00196 C
00197 C-----SHIPPING CARDS 82
00198 C -----SHIPPING CARDS 82
00199 C
00200 106 FORMAT('1',40X,' SHIPPING EMISSIONS      ',5X,I2,'/',I2,'/',I2,5X,
00201 'HR= ',I4,'00 PST'
00202 1 '40X,55(*)//10X,'CARD ECHO    KG/HR',T65,'ECHO AF
00203 2TER SPLITS IN (PPM/MIN) TIMES 1000'//10X,19('=',),T65,42('=')
00204 5   ',I J CO NOX THC
00205 3 ',T62,'I J ',T69,'CO',5X,'NO2',5X,'NO',6X,'HC1',5X,'HC2',5X,'HC
00206 43'
00207 5   '/ == == ===== == == ==
00208 3 ',T62,'= = ',T69,'==',5X,'==',5X,'==',5X,'==',5X,'=='
00209 4=')
00210 51 READ(5,101) ITYPE,II,JJ,MDATE(3),MDATE(1),MDATE(2),MHR,(C(L),L=1,
00211 1 3)
00212 IF( ITYPE.NE.82.AND. ITYPE.NE.83) GO TO 9002
00213 IF( ITYPE.EQ.83) GO TO 53
00214 ICNT = 1
00215 IF( MHR.EQ. NHR.AND. MDATE(1).EQ. NDATE(1).AND. MDATE(2).EQ. NDATE(2).
00216 1AND. MDATE(3).EQ. NDATE(3)) GO TO 52
00217 WRITE(6,102) MHR,MDATE,NHR,NDATE
00218 STOP 24
00219 52 IF(NOTE1.EQ.1) WRITE(6,106) NDATE,NHR
00220 NOTE1=2
00221 WRITE(6,104) II,JJ,(C(L),L=1, 3)
00222 C-----SPLIT NOX AND HYDROCARBONS AND REARRANGE
00223 X=0.01
00224 S1(2)=C(2)*2.0*X/(3.0-X)*CF/46.

```

```

00225      S1(3)=C(2)*2.0*(1.-X)/(3.-X)*CF/30.
00226      S1(4)=C(3)*SPLIT(1,16)*CF
00227      S1(5)=C(3)*SPLIT(2,16)*CF
00228      S1(6)=C(3)*SPLIT(3,16)*CF
00229      C-----HYDROCARBONS SPLIT
00230      S1(1)=C(1)*CF/28.
00231      C-----ECHO DATA IN PPM/MIN
00232      WRITE(6,105) II,JJ,(S1(L),L=1,6)
00233      C-----LOAD EMISSIONS INTO S(I,J,L) AND SHIFT TO INNER GRID
00234      INEW=II-MIX
00235      JNEW=JJ-MIY
00236      IF(INEW.GT.0.AND.JNEW.GT.0.AND.INEW.LE.MX.AND.JNEW.LE.MY) GO TO 59
00237      WRITE(6,406) II,JJ
00238      GO TO 51
00239      59 CONTINUE
00240      C-----LOAD INTO S(I,J,L)
00241      DO 60 L=1,6
00242      IF(S1(L).GT.0.0) S(INEW,JNEW,L)=S1(L)+S(INEW,JNEW,L)
00243      SMAX(L)=AMAX1(SMAX(L),S(INEW,JNEW,L))
00244      60 SMIN(L)=AMIN1(SMIN(L),S(INEW,JNEW,L))
00245      GO TO 51
00246      53 NOTE1 = 2
00247      IF(ICNT.EQ.0) GO TO 151
00248      1207 FORMAT(1X,T45,'SHIPPING          EMISSIONS ',4X,A4,'(PPM/MIN) ')
00249      DO 61 L=1,6
00250      WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),NHR
00251      WRITE(6,1207) NAME(L)
00252      CALL OUTPUT(S(1,1,L),MX,MY,MIX,MIY,SMAX(L),IBC)
00253      WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),NHR
00254      WRITE(6,1207) NAME(L)
00255      61 CALL PL(S(1,1,L),MX,MY,SMAX(L),SMIN(L))
00256      C-----STORE AND ZERO S
00257      DO 65 I=1,     MX
00258      DO 65 J=1,     MY
00259      DO 65 L=1,6
00260      STORE(I,J,L)=STORE(I,J,L)+S(I,J,L)
00261      65 S(I,J,L)=0.
00262      DO 67 L = 1,6
00263      SMIN(L) = 1000.
00264      67 SMAX(L) = -1.
00265      ICNT = 0
00266      GO TO 151
00267      9002 WRITE(6,9001) ITYPE
00268      STOP 72
00269      C-----RAIL DATA
00270      151 READ(5,101) ITYPE,II,JJ,MDATE(3),MDATE(1),MDATE(2),MHR,(C(L),L=1,
00271      1,3)
00272      IF(ITYPE.NE.83.AND.ITYPE.NE.84) GO TO 9004
00273      IF(ITYPE.EQ.84) GO TO 153
00274      ICNT = 1
00275      IF(MHR.EQ.NHR.AND.MDATE(1).EQ.NDATE(1).AND.MDATE(2).EQ.NDATE(2).
00276      1.AND.MDATE(3).EQ.NDATE(3)) GO TO 152
00277      WRITE(6,102) MHR,MDATE,NHR,NDATE
00278      STOP 24
00279      C-----ECHO DATA
00280      108 FORMAT('1',40X,' RAILWAY    EMISSIONS      ',5X,12,'/',12,'/',12,5X,

```

```

00281   1 'HR= ', I4, '00 PST'
00282   1           '/40X,55('*')//10X,'CARD ECHO    KG/HR',T65,'ECHO AF
00283   2TER SPLITS IN (PPM/MIN) TIMES 1000'/10X,19('='),T65,42('=')
00284   5           '/', I J CO NOX THC
00285   3 ', T62, 'I J ', T69, 'CO', 5X, 'NO2', 5X, 'NO', 6X, 'HC1', 5X, 'HC
00286   43'
00287   5           '/' == == === == == ==
00288   3 ', T62, '== = ', T69, '==', 5X, '==', 5X, '==', 6X, '==', 5X, '=='
00289   4=')
00290   152 IF(NOTE1.EQ.2) WRITE(6,108) NDATE,NHR
00291   NOTE1=3
00292   WRITE(6,104) II,JJ,C(1),C(2),C(3)
00293 C-----SPLIT AND REARRANGE NOX, HYDROCARBONS
00294   X=0.01
00295   S1(2)=C(2)*2.0*X/(3.0-X)*CF/46.
00296   S1(3)=C(2)*2.0*(1.-X)/(3.-X)*CF/30.
00297   S1(1) = C(1) *CF/28.
00298 C-----SPLIT AND REARRANGE HC
00299   S1(4)=C(3)*SPLIT(1,16)*CF
00300   S1(5)=C(3)*SPLIT(2,16)*CF
00301   S1(6)=C(3)*SPLIT(3,16)*CF
00302 C-----ECHO DATA IN PPM/MIN
00303   WRITE(6,105) II,JJ,(S1(L),L=1,6)
00304 C-----LOAD EMISSIONS INTO S(I,J,L) AND SHIFT TO INNER GRID
00305   INEW=II-MIX
00306   JNEW=JJ-MIY
00307   IF(INEW.GT.0.AND.JNEW.GT.0.AND.INEW.LE.MX.AND.JNEW.LE.MY) GO TO 15
00308   19
00309   WRITE(6,406) II,JJ
00310   GO TO 151
00311   159 CONTINUE
00312 C-----LOAD INTO S(I,J,L)
00313   DO 160 L=1,6
00314   IF(S1(L).GT.0.0) S(INEW,JNEW,L)=S1(L)+S(INEW,JNEW,L)
00315   SMAX(L)=AMAX1(SMAX(L),S(INEW,JNEW,L))
00316   160 SMIN(L)=AMIN1(SMIN(L),S(INEW,JNEW,L)).
00317   GO TO 151
00318   153 NOTE1 = 3
00319   IF(ICNT.EQ.0) GO TO 251
00320   1307 FORMAT(1X,T45,' RAIL          EMISSIONS ',4X,A4,'(PPM/MIN) ')
00321   DO 161 L=1,6
00322   WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),NHR
00323   WRITE(6,1307) NAME(L)
00324   CALL OUTPUT(S(1,1,L),MX,MY,MIX,MIY,SMAX(L),IEC)
00325   WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),NHR
00326   WRITE(6,1307) NAME(L)
00327   161 CALL PL (S(1,1,L),MX,MY,SMAX(L),SMIN(L))
00328 C-----STORE AND ZERO S
00329   DO 165 I=1, MX
00330   DO 165 J=1, MY
00331   DO 165 L=1,6
00332   STORE(I,J,L)=STORE(I,J,L)+S(I,J,L)
00333   165 S(I,J,L)=0.
00334   DO 167 L = 1,6
00335   SMIN(L) = 1000.
00336   167 SMAX(L) = -1.

```

```

00337      ICNT = 0
00338      GO TO 251
00339      9004 WRITE(6,9001) ITYPE
00340      STOP 73
00341      C-----AREA DATA
00342      251 READ(5,101) ITYPE, II, JJ, MDATE(3), MDATE(1), MDATE(2), MHR, (C(L), L=1,
00343      1 3), F(8), F(9), F(1), F(2), F(4), F(7), F(13), F(3), F(5), F(6), F(10),
00344      2 F(11), F(12)
00345      IF( ITYPE .NE. 84 .AND. ITYPE .NE. 85) GO TO 9006
00346      IF( ITYPE .EQ. 85) GO TO 253
00347      ICNT = 1
00348      IF( MHR .EQ. NHR .AND. MDATE(1) .EQ. NDATE(1) .AND. MDATE(2) .EQ. NDATE(2) ,
00349      1 AND. MDATE(3) .EQ. NDATE(3)) GO TO 252
00350      WRITE(6,102) MHR, MDATE, NHR, NDATE
00351      STOP 24
00352      109 FORMAT('1',40X,' A R E A   S O U R C E S ',5X,I2,'/',I2,'/',I2,5X,
00353      1 'HR=',I4,'00 PST'
00354      1          /40X,55('*')//10X,'CARD   ECHO    KG/HR',T74,'ECHO AF
00355      2TER SPLITS IN (PPM/MIN) TIMES 1000//10X,19('='),T74,42('=')
00356      5          /' I J   CO    NOX    THC F1 F2 F3 F4 F5 F
00357      56 F7 F8 F9 F10 F11 F12 F13',
00358      3 T74,'I J ',T81,'CO',5X,'NO',6X,'HC1',5X,'HC2',5X,'HC
00359      43'
00360      5          '/ == ==  == ==  == ==  == ==  == ==  == ==  == ==  ==
00361      3= == ==  == ==  == ==  == ==  == ==  == ==  == ==  == ==  == ==  ==
00362      3 T74,'= = ',T81,'==',5X,'==',5X,'==',6X,'==',5X,'==',5X,'==
00363      4=')
00364      252 IF(NOTE1.EQ.3) WRITE(6,109) NDATE, NHR
00365      NOTE1=4
00366      C----- TO SUPPRESS CARD ECHO ; COMMENT THE FOLLOWING WRITE
00367      C----- DO 202 N = 1,13
00368      C 202 NF(N) = IFIX(F(N)*100.)
00369      C WRITE(6,1104) II, JJ, C(1), C(2), C(3), (NF(I), I=1,13)
00370      C 1104 FORMAT(1X,I2,I2,1X,I2,F6.0,2F6.1,9I3,4I4)
00371      C-----SPLIT AND REARRANGE NOX AND HYDROCARBONS
00372      C-----NOX SPLIT
00373      X=.02
00374      S1(2)=C(2)*2.0*X/(3.-X)*CF/46.
00375      S1(3)=C(2)*2.0*(1.-X)/(3.-X)*CF/30.
00376      C----- HC SPLIT BASED SOLY ON VALUE OF THC
00377      HC1=0.
00378      HC2=0.
00379      HC3=0.
00380      DO 2541 K=1,13
00381      HC1=HC1+F(K)*SPLIT(1,K)
00382      HC2=HC2+F(K)*SPLIT(2,K)
00383      2541 HC3=HC3+F(K)*SPLIT(3,K)
00384      S1(4)=C(3)*HC1*CF
00385      S1(5)=C(3)*HC2*CF
00386      S1(6)=C(3)*HC3*CF
00387      S1(1)=C(1)*CF/28.
00388      C-----ECHO DATA IN PPM/MIN
00389      C----- SUPPRESS ECHO BY COMMENTING THE FOLLOWING WRITE

```

```

00393 C
00394 C      WRITE(6, 1055) II,JJ,(S1(L),L=1,6 )
00395 1055 FORMAT(' ',T73,I2,1X,I2,           3PF6.1,5(3PF8.3))
00396 C-----LOAD EMISSIONS INTO S AND SHIFT TO INNER GRID
00397      INEW=II-MIX
00398      JNEW=JJ-MIY
00399      IF(INEW.GT.0.AND.JNEW.GT.0.AND.INEW.LE.MX.AND.JNEW.LE.MY) GO TO 25
00400      19
00401 C
00402 C - - - - SUPRESS PRINTS OF EXTERNAL SOURCES
00403 C
00404 C      WRITE(6,406) II,JJ
00405      GO TO 251
00406 259 CONTINUE
00407 C-----LOAD INTO S(I,J,L)
00408      DO 260 L=1,6
00409      IF(S1(L).GT.0.0) S(I,NEW,J,NEW,L)=S1(L)+S(I,NEW,J,NEW,L)
00410      SMAX(L)=AMAX1(SMAX(L),S(I,NEW,J,NEW,L))
00411 260 SMIN(L)=AMIN1(SMIN(L),S(I,NEW,J,NEW,L))
00412      GO TO 251
00413 253 NOTE1 = 4
00414      IF(ICNT.EQ.0) GO TO 351
00415      DO 261 L=1,6
00416      WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),MHR
00417 1407 FORMAT(1X,T45,' AREA          EMISSIONS ',4X,A4,'(PPM/MIN) ')
00418      WRITE(6,1407) NAME(L)
00419      CALL OUTPUT(S(1,1,L),MX,MY,MIX,MIY,SMAX(L),IBC)
00420      WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),MHR
00421      WRITE(6,1407) NAME(L)
00422 261 CALL PLC(S(1,1,L),MX,MY,SMAX(L),SMIN(L))
00423 C-----STORE
00424      DO 265 I=1, MX
00425      DO 265 J=1, MY
00426      DO 265 L=1,6
00427      STORE(I,J,L)=STORE(I,J,L)+S(I,J,L)
00428 265 S(I,J,L)=0.
00429      DO 267 L = 1,6
00430      SMIN(L) = 1000.
00431 267 SMAX(L) = -1.
00432      ICNT = 0
00433      GO TO 351
00434 9006 WRITE(6,9001) ITYPE
00435      STOP 74
00436 C-----POINT SOURCES WITH NO STACK (GROUND LEVEL POINT SOURCE)
00437 351 READ(5,101) ITYPE,II,JJ,MDATE(3),MDATE(1),MDATE(2),MHR,(C(L),L=1,
00438      1 3),F(8),F(9),F(1),F(2),F(4),F(7),F(13),F(3),F(5),F(6),F(10),
00439      2 F(11),F(12)
00440      IF(ITYPE.NE.85.AND.ITYPE.NE.91) GO TO 9008
00441      IF(ITYPE.EQ.91) GO TO 353
00442      ICNT = 1
00443      IF(MHR.EQ.MHR.AND.MDATE(1).EQ.NDATE(1).AND.MDATE(2).EQ.NDATE(2).
00444      1 AND MDATE(3).EQ.NDATE(3)) GO TO 352
00445      WRITE(6,102) MHR,MDATE,NHR,NDATE
00446      STOP 24
00447 110 FORMAT('!',40X,'GROUND LEVEL POINT SOURCE',5X,I2,'/',I2,'/',I2,5X,
00448      1'HR=',I4,'00 PST'

```

```

00449      1           /40X,55('*')//10X,'CARD ECHO   KG/HR',T74,'ECHO AF
00450      2TER SPLITS IN (PPM/MIN) TIMES 1000'/10X,19('='),T74,42('=')
00451      5           /' I J CO NOX THC F1 F2 F3 F4 F5 F
00452      56 F7 F8 F9 F10 F11 F12 F13',
00453      3   T74,'I J ',TB1,'CO',5X,'NO2',5X,'NO',6X,'HC1',5X,'HC2',5X,'HC
00454      43'.
00455      5           /' == == === == == == == == == == == ==
00456      3= == == == == == == == == == == == == == == == ==
00457      3   T74,'= = ',TB1,'==',5X,'==',5X,'==',5X,'==',5X,'==
00458      4= ')
00459      352 IF( NOTE1.EQ.4) WRITE(6,110) NDATE , NHR
00460      NOTE1=5
00461 C
00462 C - - - - SUPPRESS CARD ECHO ; COMMENT THE FOLLOWING WRITE
00463 C
00464 C DO 303 N = 1.13
00465 C 303 NF(N) = IFIX(F(N)*100.)
00466 C WRITE(6,1104) II,JJ,C(1),C(2),C(3),(NF(I),I=1,13)
00467 C-----SPLIT AND REARRANGE
00468 C-----NOX SPLIT
00469 X=0.02
00470 S1(2)=C(2)*2.0*X/(3.-X)*CF/46.
00471 S1(3)=C(2)*2.0*(1.-X)/(3.-X)*CF/30.
00472 C----- HC SPLIT BASED ON THC ONLY
00473 HC1=0.
00474 HC2=0.
00475 HC3=0.
00476 DO 3541 K=1,13
00477 HC1=HC1+F(K)*SPLIT(1,K)
00478 HC2=HC2+F(K)*SPLIT(2,K)
00479 HC3=HC3+F(K)*SPLIT(3,K)
00480 3541 CONTINUE
00481 S1(1)=C(1)*CF/28.
00482 C
00483 C-----SPLIT AND REARRANGE HC
00484 C
00485 S1(4)=C(3)* HC1      *CF
00486 S1(5)=C(3)* HC2      *CF
00487 S1(6)=C(3)* HC3      *CF
00488 C-----ECHO DATA IN PPM/MIN
00489 C
00490 C - - - - SUPPRESS ECHO BY COMMENTING THE FOLLOWING WRITE
00491 C
00492 C      WRITE(6,1055) II,JJ,(S1(L),L=1,6)
00493 C-----LOAD EMISSIONS INTO S AND SHIFT TO INNER GRID
00494     INEW= II-MIX
00495     JNEW= JJ-MIX
00496     IF( INEW.GT.0.AND.JNEW.GT.0.AND.INEW.LE.MX.AND.JNEW.LE.MY) GO TO 35
00497     19
00498 C
00499 C - - - - SUPPRESS PRINTS OF EXTERNAL SOURCES
00500 C
00501 C      WRITE(6,406) II,JJ
00502 C      GO TO 351
00503 359 CONTINUE
00504 C-----LOAD INTO S(I,J,L)

```



```

00561      JNEW=JJ-MIY
00562      IF( INEW.GT.0 .AND. JNEW.GT.0 .AND. INEW.LE.MX .AND. JNEW.LE.MY) GO TO 459
00563      WRITE(6,406) II,JJ
00564      GO TO 451
00565      459 CONTINUE
00566      C ****
00567      SPEED = WIND(INEW,JNEW)
00568      STAB = FLOAT(KSTAB(INEW,JNEW))
00569      C ****
00570      C----DO PLUME RISE
00571      CALL PLUMER(IFLGG,STKHT,TPLUME,VOLFLO,1.E6,SPEED,STAB,2,10.,
00572      1          0.,0.,0.,0.,0.,0.,0.,RISE)
00573      TOTRIS = RISE + STKHT
00574      DO 410 K = 1,MZ
00575      IF(TOTRIS.GT.HT(K)) GO TO 410
00576      KK = K
00577      CFK = CF*DZ(1)/DZ(K)
00578      GO TO 454
00579      410 CONTINUE
00580      GO TO 451
00581      454 CONTINUE
00582      C----CONVERT TO PPM/MIN, THE RESOLUTION \Z IS DZ(K)
00583      C----REARRANGE AND SPLIT
00584      C----HC SPLIT
00585      C----REARRANGE
00586      X = .05
00587      S1(1)=C(1)*CFK/28.
00588      S1(2)=C(2)*2.0*X/(3.0-X)*CFK/46.
00589      S1(3)=C(2)*2.0*(1.-X)/(3.-X)*CFK/30.
00590      S1(4)=C(3)*SPLIT(1,16)*CFK
00591      S1(5)=C(3)*SPLIT(2,16)*CFK
00592      S1(6)=C(3)*SPLIT(3,16)*CFK
00593      C----ECHO DATA IN PPM/MIN
00594      1052 FORMAT(' ',T61,I2,1X,I2,     3PF6.1,5(3PF8.3),2X,0PF5.0,2X,I2)
00595      WRITE(6,1052)II,JJ, (S1(L),L=1,6),RISE,KK
00596      C----LOAD EMISSIONS INTO S AND SHIFT TO INNER GRID
00597      C----LOAD INTO SE(L,I,J,K)
00598      K-KK
00599      DO 460 L=1,6
00600      460 SE(L,INEW,JNEW,K)=S1(L)+SE(L,INEW,JNEW,K)
00601      GO TO 451
00602      453 NOTE1 = 6
00603      IF(ICNT.EQ.0) GO TO 651
00604      C * -----614
00605      C SUPPRESS 3 D PRINTS
00606      C -----L 5340
00607      GO TO 651
00608      DO 472 K=1,MZ
00609      DO 472 L=1,6
00610      SMAX1 = -1.
00611      DO 471 I=1,MX
00612      DO 471 J=1,MY
00613      S(I,J,L)=SE(L,I,J,K)
00614      471 SMAX1 = AMAX1(SMAX1,S(I,J,L))
00615      WRITE(6,125) NDATE(1),NDATE(2),NDATE(3),NAME(L),K
00616      472 CALL OUTPUT(S(1,1,L),MX,MY,MIX,MIY,SMAX1,IBC)

```



```

00673      654 CONTINUE
00674 C-----NOX SPLIT AND HYDROCARBONS SPLIT AND REARRANGE
00675      X=0.01
00676      S1(1)=C(1)*CFK/28.
00677      S1(2)=C(2)*2.*X/(3.-X)*CFK/46.
00678      S1(3)=C(2)*2.*((1.-X)/(3.-X))*CFK/30.
00679 C-----HC SPLIT
00680      S1(4)=C(3)*(F1*SPLIT(1,17) + F2*SPLIT(1,18))*CFK
00681      S1(5)=C(3)*(F1*SPLIT(2,17) + F2*SPLIT(2,18))*CFK
00682      S1(6)=C(3)*(F1*SPLIT(3,17) + F2*SPLIT(3,18))*CFK
00683      1054 FORMAT(1X, T61, I2, 1X, I2, 3PF6.1, 5(3PF8.3), 2X, I2)
00684      WRITE(6, 1054) II, JJ, (S1(L), L=1, 6), KK
00685 C-----LOAD EMISSIONS INTO SE AND SHIFT TO INNER GRID
00686      INEW=II-MIX
00687      JNEW=JJ-MIY
00688      IF( INEW.GT.0 .AND. JNEW.GT.0 .AND. INEW.LE.MX .AND. JNEW.LE.MY) GO TO 659
00689      WRITE(6, 406) II, JJ
00690      GO TO 651
00691      659 CONTINUE
00692      DO 660 L=1, 6
00693      660 SE(L, INEW, JNEW, KKO)=S1(L)+SE(L, INEW, JNEW, KKO)
00694      GO TO 651
00695      653 IF( ICNT .EQ. 0) GO TO 751
00696 C * -----614
00697 C     SUPRESS   3 D    PRINTS
00698 C     -----L 5340
00699      GO TO 751
00700      DO 672 K=1, MZ
00701      DO 672 L=1, 6
00702      SMAX1 = -1.
00703      DO 671 I=1, MX
00704      DO 671 J=1, MY
00705      S(I, J, L)=SE(L, I, J, K)
00706      671 SMAX1 = AMAX1(SMAX1, S(I, J, L))
00707      WRITE(6, 127) NDATE(1), NDATE(2), NDATE(3), NAME(L), K
00708      127 FORMAT('1', I2, '/', I2, '/', I2, 5X, 'ELEVATED SOURCES(STACKS+AIRCRAFT',
00709      1, 5X, A4, 'PPM/MIN', 5X, 'K= ', I2)
00710      672 CALL OUTPUT(S(1, 1, L), MX, MY, MIX, MIY, SMAX1, IBC)
00711      GO TO 751
00712      9020 WRITE(6, 9001) ITYPE
00713      STOP 24
00714      9001 FORMAT('1', 'OUTPUT FROM PRESOR''0', 18('*')//'*0', 'CODE STOPPED
00715      1BECAUSE CARD IMAGE TYPES NOT IN CORRECT ORDER. THE CARD IMAGE
00716      2WHICH STOP THE PROGRAM WAS TYPE= ', I2)
00717      9009 FORMAT('1', I2, '/', I2, '/', I2, 5X, 'POINT SOURCE(NO STACK) EMISSIONS',
00718      15X, A4, 'PPM/MIN')
00719      751 CONTINUE
00720      DO 752 J = 1, MY
00721      DO 752 I = 1, MX
00722      DO 752 L = 1, 6
00723      SE(L, I, J, 1) = SE(L, I, J, 1) + STORE(I, J, L)
00724      S(I, J, L) = SE(L, I, J, 1)
00725      SMAX(L) = AMAX1(SMAX(L), S(I, J, L))
00726      752 SMIN(L) = AMIN1(SMIN(L), S(I, J, L))
00727 C-----PRINT AND PLOT TOTAL SOURCE
00728      1907 FORMAT(/1X, T45, '** ALL SOURCES LEVEL 1', 4X, A4, '(PPM/MIN')/

```

```
00729      DO 754 L=1,6
00730      WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),NHR
00731      WRITE(6,1907) NAME(L)
00732      CALL OUTPUT(S(1,1,L),MX,MY,MIX,MIY,SMAX(L),IEC)
00733      WRITE(6,107) NDATE(1),NDATE(2),NDATE(3),NHR
00734      WRITE(6,1907) NAME(L)
00735 754   CALL PLC(S(1,1,L),MX,MY,SMAX(L),SMIN(L))
00736 C ****
00737 C -----FIRST RECORD IS THE HEADER
00738 C-----SECOND RECORD IS SE
00739      WRITE(12) NDATE ,NHR
00740      WRITE(12) (((SE(L,I,J,K) .L=1,6) ,I=1,MX) ,J=1,MY) ,K=1,MZ)
00741      C ****
00742      C ****
00743 C ****
00744 C ****
00745      DO 767 L = 1,6
00746      SMIN(L) = 1000.
00747 767   SMAX(L) = -1.
00748      RETURN
00749      END
```

```

00001      SUBROUTINE PLUMER( IFLG, HS, TE, VOL, R, WIND, STAB, IUR, HTMET, T1, T2, T3,
00002          2           T4, Z1, Z2, Z3, Z4, DH)
00003      C
00004      C      THIS PROGRAM CALCULATES THE PLUME RISE AT THE DOWNWIND DISTANCE R.
00005      C      EITHER BRIGGS'67, BRIGGS'74, TVA OR A SPECIFIED PLUME RISE CAN BE
00006      C      USED.
00007      C
00008      DIMENSION TG(6),P(6)
00009      DATA TG/- .019,- .018,- .016,- .01,0.,0.015/
00010      DATA P / .15,.17,.20,.26,.39,.48/
00011      C
00012      C      CALCULATE VELOCITY AT TOP OF STACK
00013      C
00014      C      WRITE(6,800) IFLG, HS, TE, VOL, R, WIND, STAB, IUR, HTMET, T1, T2, T3,
00015          1           T4, Z1, Z2, Z3, Z4, UDH, DH
00016      C800  FORMAT(1X, 10G11.2)
00017      IS = IFIX(STAB+.49)
00018      U=WIND*( HS/HTMET)**P( IS)
00019      IF( IFLG.GT.3) GO TO 700
00020      C
00021      C      COMPUTE TEMPERATURE GRADIENT AND STABILITY
00022      C
00023      IF( HS.GT.Z4) GO TO 30
00024      IF( HS.LT.Z4.AND.HS.GT.Z3) GO TO 40
00025      IF( HS.LT.Z3.AND.HS.GT.Z2) GO TO 50
00026          GT=(T2-T1)/(Z2-Z1)
00027          TO = T1 + GT *(HS-Z1)
00028          GO TO 60
00029      30  TO = T1
00030          IF( T1.EQ.0.0) TO = 27.0
00031          GT= TG( IS)
00032          GO TO 60
00033      40  GT=(T4-T3)/(Z4-Z3)
00034          TO = T3 + GT *(HS-Z3)
00035          GO TO 60
00036      50  GT = (T3-T2)/(Z3-Z2)
00037          TO = T2 + GT *(HS-Z2)
00038          60  TA = 273.+TO
00039      C
00040      C      CALCULATE BUOYANCY F
00041      C
00042          F= 9.8*VOL*(( TE+273.)-TA)/( TE+273.)/3.14159
00043      C
00044          IF( IFLG.GT.2) GO TO 300
00045          IF( IFLG.EQ.2) GO TO 200
00046      C
00047      C      BRIGGS 1967
00048      C
00049          IF( IS.GT.4) GO TO 101
00050          DH=1.6*F**((1./3.)*(AMIN1(R,10.*HS))**((2./3.)/U
00051          GO TO 999
00052      101  IF( GT.LT.-.005) GT=-.005
00053          DH=2.9*(F/(U*9.8*(GT+0.01)/TA))**((1./3.))
00054          GO TO 999
00055      C
00056      C      BRIGGS 1974

```

```

00057      C
00058      200 IF( IS.GT.4) GO TO 201
00059          IF(F.GT.55.) XX= 34.* F**0.4
00060          IF(F.LE.55.) XX = 14.*F**0.625
00061          DH=1.6*F**(.1./3.)*(AMIN1(R,XX*3.5)**(2./3.))/U
00062          GO TO 999
00063      201 IF(GT.LT.-.005) GT=-.005
00064          DH=2.6*(F/(U*9.8*(GT+0.01)/TA))**(1./3.)
00065          GO TO 999
00066      C
00067      C      TVA
00068      C
00069      300 IF( IS.LT. 4) GO TO 200
00070          IF(GT.LT.-.017) GO TO 200
00071          IF(R.GT. 1824) GO TO 310
00072          IF(GT.GE.-.0084) GO TO 301
00073      C      NEUTRAL
00074          DH=2.50*F**(.1./3.)*R**0.56/U
00075          GO TO 999
00076      C      MODERATELY STABLE
00077      301 IF(GT.GE.-.0030) GO TO 302
00078          DH=3.75*F**(.333333)*R**0.49/U
00079          GO TO 999
00080      C      VERY STABLE
00081      302 DH=13.8 *F**(.1./3.)*R**(.26)/U
00082          GO TO 999
00083      C      R GREATER THAN 1824 METERS
00084      310 DH=173.*F**(.1./3.)*EXP(-64.*((GT+0.01))/U
00085          GO TO 999
00086      700 CONTINUE
00087      IF(IFLG.EQ.3) DH=UDH
00088      999 CONTINUE
00089      RETURN
00090      END

```

```

00001      SUBROUTINE OUTPUT(ARRAY, MX, MY, MIX, MIY, XMAX, IBC)
00002      C
00003      C ..... OUTPUT IS CALLED TO PRINT ANY MX*MY ARRAY
00004      C ..... USING AN OFFSET OF MIX AND MIY AND GIVEN THE
00005      C ..... MAXIMUM VALUE IN THE ARRAY = XMAX (WHICH MUST BE POSITIVE)
00006      C
00007      C
00008      C ..... PRINT THE EULERIAN CONCENTRATIONS
00009      C     DIMENSION ICE( 50),           ARRAY(MX, MY), IBC(MY)
00010      C
00011      C ..... FIND FACTOR FOR A TWO DIGIT PRINT .....
00012      C
00013      F = 1.
00014      FACTOR = 1.
00015      IF(XMAX.LT.1.E-30) GO TO 11
00016      LOGX = IFIX(ALOG10(XMAX))
00017      IF(XMAX .LT. 1.0) LOGX = LOGX - 1
00018      FACTOR = 10.**(LOGX - 1)
00019      F = 1./FACTOR
00020      11  CONTINUE
00021      102  FORMAT('0', T47, 'MULTIPLY VALUES IN THIS TABLE BY', 1PE8.1)
00022      WRITE(6, 102) FACTOR
00023      MOS=MIX+1
00024      MOT=MX+MIX
00025      103  FORMAT(' ',5X,42(I3))
00026      WRITE(6, 103) (I,I=MOS,MOT)
00027      WRITE(6, 103) (I,I = 1, MX)
00028      C
00029      C ..... PRINT CONCENTRATIONS .....
00030      C
00031      DO 30 JJ = 1, MY
00032      J = MY - JJ + 1
00033      JL=J+MIY
00034      LX = IBC(J)
00035      DO 15 I = 1, LX
00036      A = ARRAY(I, J)*F + 0.5
00037      IF(A.LT.0.5) A = A - 1.0
00038      15   ICE(I) = IFIX(A)
00039      104  FORMAT('0', I2, 1X, I2, 42(I3))
00040      WRITE(6, 104) JL, J, (ICE(I), I = 1, LX)
00041      30   CONTINUE
00042      RETURN
00043      END

```

```

00001 C
00002 SUBROUTINE PL(Z,NX,NY,ZMAX,ZMIN)
00003 ****
00004 C*
00005 C* P L IS A PRINTER-PLOTTER ROUTINE WHICH *
00006 C* SCALES MATRIX VALUES FROM 0 TO 9 *
00007 C*
00008 ****
00009 C
00010 DIMENSION LQ(72),L(131),Z(NX ,NY)
00011 DATA NNC,AMAX,AMIN,LOGFL/10,0.,0.,0/
00012 DATA LQ /'0','1','2','3','4','5',
00013 .,'6','7','8','9','A','B',
00014 .,'C','D','E','F','G','H',
00015 .,'J','K','L','M','N','O',
00016 .,'P','Q','R','S','T','U',
00017 .,'W','X','Y','Z','/,
00018 DATA ISTAR/*/,IMAX/60/
00019 C
00020 C
00021 C
00022 LOGFL = 0
00023 AMIN = 0.0
00024 AMAX = 0.0
00025 NNC = 10
00026 NXMX=NX
00027 C ASPECT = DY/DX
00028 ASPECT = 1.0
00029 JMAX = .6*NY/NX*IMAX*ASPECT +.5
00030 NC = MIN0(NNC,36)
00031 ZMIN = AMIN
00032 ZMAX = AMAX
00033 ANX = NX
00034 ANY = NY
00035 IF (ABS(ZMAX-ZMIN) .GT. 1.0E-5) GO TO 40
00036 ZMAX = Z(1,1)
00037 ZMIN = ZMAX
00038 DO 20 I=1,NX
00039 DO 20 J=1,NY
00040 ZMAX = AMAX1(ZMAX,Z(I,J))
00041 20 ZMIN = AMIN1(ZMIN,Z(I,J))
00042 IF (ABS(ZMAX-ZMIN) .LT. 1.0E-20) RETURN
00043 40 IF (LOGFL .EQ. 0) GO TO 80
00044 IF (ZMIN .GT. 0.) GO TO 60
00045 WRITE (6,220)
00046 RETURN
00047 60 HZMAX = ZMAX
00048 HZMIN = ZMIN
00049 ZMAX = ALOG10(ZMAX)
00050 ZMIN = ALOG10(ZMIN)
00051 80 SC = (NC*2.-1.E-5)/(ZMAX-ZMIN)
00052 DX = ANX/IMAX
00053 DY = ANY/JMAX
00054 Y = ANY+DY
00055 DO 180 JJ=1,JMAX
00056 J = JMAX -JJ +1

```

```

00057      Y = Y-DY
00058      M = Y + .5
00059      DM = AMAX1(Y - M + .5, 1.0E-10)
00060      IF(M.EQ.0) DM = 1.0
00061      IF(M.GE.NY) DM = 0.0
00062      X = 0.
00063      DO 160 I=1,IMAX
00064      X = X+DX
00065      IF (I*NJ.EQ. 1.OR. I*NJ.EQ. IMAX*JMAX) GO TO 140
00066      IF (I .EQ. 1 .AND. J .EQ. JMAX .OR. J .EQ. 1 .AND. I .EQ. IMAX)
00067          GO TO 140
00068      N = X + .5
00069      DN = AMAX1(X - N + .5, 1.0E-10)
00070      IF(N.LE.0) DN = 1.0
00071      IF(N.GE.NX) DN = 0.0
00072      IF (LOGFL .NE. 0) GO TO 100
00073      DMM = AMAX1(1. - DM, 1.0E-10)
00074      DNN = AMAX1(1. - DN, 1.0E-10)
00075      IF(M.EQ.0) DMM = 0.0
00076      IF(M.LE.0) M=1
00077      IF(N.EQ.0) DNN = 0.0
00078      IF(N.LE.0) N=1
00079      NP1=N+1
00080      IF(NP1.GT.NX) NP1=NX
00081      MP1=M+1
00082      IF(MP1.GT.NY) MP1=NY
00083      C = DN*(DM*Z(NP1,MP1)+DMM*Z(NP1,M))+DNN*(DM*Z(N,MP1)+DMM*
00084      .Z(N,M))
00085      GO TO 120
00086      100 C = DN*(DM* ALOG10(Z(N+1,M+1))+(1.-DM)* ALOG10(Z(N+1,M))+ (1.-DN)*
00087      .(DM* ALOG10(Z(N,M+1))+(1.-DM)* ALOG10(Z(N,M)))
00088      120 IND = 1+SC*(C-ZMIN)
00089      IF (IND .LE. 0) IND = 1
00090      IF (IND .GT. 2*NC) IND = 2*NC
00091      L(I) = LQ(IND)
00092      GO TO 160
00093      140 L(I) = ISTAR
00094      160 CONTINUE
00095
00096      C      INDEX CHANGE TO MAKE PRINTOUT IN MATRIX - LIKE FORM
00097      C
00098      180 WRITE (6,240) (L(I) , I = 1 , IMAX)
00099      IF (LOGFL .EQ. 0) GO TO 200
00100      ZMAX = HZMAX
00101      ZMIN = HZMIN
00102      200 WRITE (6,260) ZMIN,ZMAX
00103
00104      C      RETURN
00105      220 FORMAT (/////
00106      .      115H ****NON-POSITIVE VALUE IN ARRAY TO BE
00107      . CONTOURED LOGARITHMICALLY. PL IS RETURNING WITHOUT PLOTTING )
00108      C
00109      240 FORMAT (1X,131A1)
00110      260 FORMAT (/17H0MINIMUM VALUE = ,1PE9.3,5X,16HMAXIMUM VALUE = ,E9.3)
00111      280 FORMAT(1H1)
00112      END

```

A.3.4 MADCAP

```

00001 C THIS IS MADCAP DEVELOPED BY R. C. SKLAREW, J. C. WILSON
00002 C R. J. GELINAS & A. J. FABRICK
00003 C VERSION 020202020202 7/23/76
00004 C COMMON NVERS(6),NDCOD1,NDCOD2,NDCOD3,NPLACE(4),NDATE(3),NHR,NOHRS
00005 C 1 , MX, MY, MZ, MS, NS, DX, DY, NSTEP, AVESOL, UF, UFX, DI, LI, LIS, UFY
00006 C 1 , INT1, INT, MXY, MXYZ, MXYZS, MXS, MYS, MXYS, MXZ, MYZ, MXZS, MYZS
00007 C 1 , DZ(MZ), CEG(MX, MY, MZ, NS), SE(MS, MX, MY, MZ)
00008 C 1 , IBC(MY), BCX(NS, MY, MZ, 2), BCY(NS, MX, MZ, 2), SOL(MX, MY)
00009 C 1 , CEG(MX, MY, MS), BCXG(MS, MY, 2), BCYG(MS, MX, 2)
00010 C 1 , U(MX, MY, MZ, 2), D(MX, MY, MZ), H2O(MX, MY), NSPEC(2, MS)
00011 C 1 , IPRTSP( MS)
00012 C 1 , UFZ2(MZ) , XMAX( NS, 2 ), SL( MS, INT )
00013 C 1 , CC( NS ), CL(MX, MY, MZ, NS ), XMIN(NS), ICOAST(MY)
00014 C COMMON NVERS(6),NDCOD1,NDCOD2,NDCOD3,NPLACE(4),NDATE(3),NHR,NOHRS
00015 C 1 , MX, MY, MZ, MS, NS, DX, DY, NSTEP, AVESOL, UF, UFX, DI, LI, LIS, UFY
00016 C 1 , INT1, INT, MXY, MXYZ, MXYZS, MXS, MYS, MXYS, MXZ, MYZ, MXZS, MYZS
00017 C 1 , DZ(05), CE(37, 51, 05, 13), SE(07, 37, 51, 05)
00018 C 1 , IBC(51), BCX(13, 51, 05, 02), BCY(13, 37, 05, 2), SOL(37, 51)
00019 C 1 , CEG(37, 51, 07), BCXG(07, 51, 2), BCYG(07, 37, 2)
00020 C 1 , U(37, 51, 05, 2), D(37, 51, 05), H2O(37, 51), NSPEC(2, 07)
00021 C 1 , IPRTSP( 07)
00022 C 1 , UFZ2(05) , XMAX(13, 2 ), SL( 07, 11 )
00023 C 1 , CC( 13 ), CL(37, 51, 05, 13), XMIN(13), ICOAST(51)
00024 C DIMENSION NVERS1(6), NSPEC1(2, MS), DZ1(MZ)
00025 C DIMENSION NVERS1(6), NSPEC1(2, 07), DZ1(05), CE1(1), IBC1(60)
00026 C INTEGER IC1(60), IX(5), JY(5)
00027 C REAL*8 PLACE(5)
00028 C DIMENSION STOR(11, 5, 5, 7)
00029 C NAMELIST/TERR0/IBC1, IC1
00030 C NAMELIST/GRID0/MX1, MY1, MZ1, MIX, MIY
00031 C EQUIVALENCE (CE1, CE)
00032 C*****
00033 C*****SITE SPECIFIC DATA*****
00034 C DATA PLACE / ' KEARNY ' , ' DIEGO ' ,
00035 C . , ' SANTEE ' , ' EL CAJON '
00036 C . , ' CHOLLAS ' /
00037 C DATA IX/25, 23, 32, 32, 28/, JY/22, 15, 22, 20, 16/
00038 C DATA NOPL/5/, MYDSK/1/, NEDIT/1/
00039 C DATA DX1, DY1, DZ1/2*2, E3, 50., 75., 125., 250., 500./
00040 C*****
00041 C DATA NVERS1/02, 02, 02, 02, 02, 02/, INIT/1/, NSPEC1/' C ', ' 0 '
00042 C 1 , ' OZ ', ' ONE ', ' NO ', ' 2 ', ' N ', ' 0 ', ' HC ', ' 1 '
00043 C 1 , ' HC ', ' 2 ', ' HC ', ' 3 '
00044 C DATA NOTE/1/, NS1/13/, MS1/7/
00045 C READ(5, GRID0)
00046 C READ(5, TERR0)
00047 C DO 1 I = 1, 6
00048 C 1 NVERS(I) = NVERS1(I)
00049 C MX = MX1
00050 C MY = MY1
00051 C MZ = MZ1
00052 C MS = MS1
00053 C NS = NS1
00054 C DX = DX1
00055 C DY = DY1
00056 C INT= INIT

```

```

00057      INT1= INIT-1
00058      MXYZ = MX*MY*MZ
00059      MNXYZS = MXYZ*MS + 1
00060      NXYZS = NS*MXYZ
00061      DO 11 N = MNXYZS, NXYZS
00062      11 CE1(N) = 0.
00063      DO 27 NN=1, MY
00064      ICOAST(NN)=MAX0( IC1(NN+MIY)-MIX, 1)
00065      27 IBC(NN)= MIN0( IBC1(NN+ MIY)- MIX, MX)
00066      DO 2 I = 1, MZ
00067      2 DZ(I) = DZ1(I)
00068      DO 3 I = 1, MS
00069      NSPEC(1, I) = NSPEC1(1, I)
00070      3 NSPEC(2, I) = NSPEC1(2, I)
00071      CALL INITIAL
00072      NRR=NHR
00073      CALL OUTPUT(MYDSK, NEDIT)
00074      DO 10 IHR= 1, NOHRS
00075      CALL SETUP
00076      CALL TRACE
00077      NHR=NHR+1
00078      CALL OUTPUT(MYDSK, NEDIT)
00079      DO 300 L = 1, MS
00080      DO 300 K = 1, MZ
00081      DO 300 N = 1, NOPL
00082      NIX= IX(N)
00083      NJY= JY(N)
00084      300 STOR(IHR, N, K, L) = CE(NIX, NJY, K, L)
00085      10 CONTINUE
00086      NST=NRR+NOHRS
00087      NRR=NRR+1
00088      DO 303 L = 1, MS
00089      WRITE(6,313) NSPEC(1, L), NSPEC(2, L)
00090      313 FORMAT('1', T60, 2A4)
00091      WRITE(6, 314) (N, N=NRR, NST)
00092      314 FORMAT(/1X, 'STATION ', 11(4X, I2, 4X))
00093      DO 301 NSITE = 1, 5
00094      301 WRITE(6, 315) PLACE(NSITE), (STOR(IHR, NSITE, 1, L), IHR=1, NOHRS)
00095      315 FORMAT(/5X, A8 ,6X, 11(1X, F7.3, 2X), 2(/, 19X, 11(1X, F7.3, 2X)))
00096      DO 302 NSITE = 1, 5
00097      WRITE(6, 316)
00098      316 FORMAT(///)
00099      DO 302 K = 1, MZ
00100      KK = MZ - K + 1
00101      302 WRITE(6, 317) PLACE(NSITE), KK, (STOR(IHR, NSITE, KK, L), IHR=1, NOHRS)
00102      317 FORMAT(5X, A8, 3X, I2, 1X, 11(1X, F7.3, 2X), 2(/, 19X, 11(1X, F7.3, 2X)))
00103      303 CONTINUE
00104      STOP
00105      END

```

```

00001      SUBROUTINE INITIAL
00002      COMMON NVERS(6), NDCOD1, NDCOD2, NDCOD3, NPLACE(4), NDATE(3), NHR, NOHRS
00003      1      , MX, MY, MZ, MS, NS, DX, DY, NSTEP, AVESOL, UF, UFX, DI, LI, LIS, UFY
00004      1      , INT1, INT, MXY, MXYZ, MXYZS, MXS, MYS, MXYS, MXZ, MYZ, MXZS, MYZS
00005      1      , DZ(05), CE(37,51,05,13), SE(07,37,51,05)
00006      1      , IBC(51), BCX(13,51,05,02), BCY(13,37,05,2), SOL(37,51)
00007      1      , CEG(37,51,07), BCXG(07,51,2), BCYG(07,37,2)
00008      1      , U(37,51,05,2), D(37,51,05), H20(37,51), NSPEC(2,07)
00009      1      , IPRTSP( 07)
00010      1      , UFZ2(05)      , XMAX(13,2), SL( 07,11)
00011      1      , CC( 13), CL(37,51,05,13), XMIN(13), ICOAST(51)
00012      DIMENSION CE1(1), MDATE(3), LIT(2)
00013      EQUIVALENCE (CE1,CE)
00014      DATA ITEST//',LIT//' NO ', ' YES'
00015      READ(5,100) NPLACE, NOHRS, NHR, NDATE, IPRTSP
00016      100 FORMAT(4A4,(40I2))
00017      PRINT 101,NPLACE,NVERS,NOHRS,NHR,NDATE,( NSPEC(1,N), NSPEC(2,N)
00018      1      , LIT(1 + IPRTSP( N)), N = 1, MS)
00019      1      , LIT(1 + IPRTSP( N)), N = 1, NS)
00020      101 FORMAT(1H1, 44X,4A4,/,40X, 'MADCAP VERSION',6I3,/,5X,
00021      1      'DEVELOPED BY R. C. SKLAREW, J. C. WILSON, R. J. CELINAS & ',
00022      1      'A. J. FABRICK'//5X, 'CALCULATION FOR' ,13, ' HOURS STARTING',I3,
00023      1      '00PST',
00024      1      ' ON ',I2,'/',I2,'/',I2   // 5X, 'THE FOLLOWING SPECIES ARE TO BE'
00025      1      ' PRINTED: '//(10X,2A4,5X,A4))
00026      102 FORMAT(//10X, 'G R I D P A R A M E T E R S'
00027      1      10X, '=====//9X, 'MX =' ,I3,4X, 'DX'
00028      2      ' =' ,F6.0, ' METERS'//9X, 'MY =' ,I3,4X, 'DY =' ,F6.0//9X,
00029      3      'MZ =' ,I3,4X, 'DZ(K)' = ,3F6.0,7F7.0)
00030      WRITE(6,102) MX,DX,MY,DY,MZ,DZ
00031      MXY=MX*MY
00032      MXYZ=MXY*MZ
00033      MXYZS=MXYZ*MS
00034      MXS=MX*MS
00035      MYS=MY*MS
00036      MXYS=MXS*MY
00037      MXZ=MX*MZ
00038      MYZ=MY*MZ
00039      MYZS=MYZ*MS
00040      MXZS=MXZ*MS
00041      DI=60./INT1
00042      LI=MS*INT1
00043      LIS=LI+MS
00044      UF=1./FLOAT(INT1)
00045      UFX=UF*3600./DX
00046      UFY=UF*3600./DY
00047      DO 7 K=1,MZ
00048      UFZ2(K)=3600./(DZ(K)**2)
00049      7      CONTINUE
00050      C ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::
00051      C :: IF( ITEST.LT.1) GO TO 40
00052      C :: ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::
00053      C :: DO 10 IU = 10,12
00054      8      READ(IU) MDATE, MHR
00055      C :: IF(MDATE(1).EQ.NDATE(1).AND.MDATE(2).EQ.NDATE(2).AND.MDATE(3).
00056      C :: EQ.NDATE(3).AND.MHR.EQ.NHR) GO TO 10

```

```

00057 IF(MDATE(1).GT.NDATE(1)) STOP 12
00058 IF(MDATE(2).GT.NDATE(2)) STOP 30
00059 IF(MDATE(3).GT.NDATE(3)) STOP 1675
00060 IF(MHR.GT.NHR.AND.MDATE(2).EQ.NDATE(2)) STOP 24
00061 READ(IU)
00062 IF(IU.EQ.12) GO TO 8
00063 READ(IU)
00064 IF(IU.EQ.10) GO TO 8
00065 READ(IU)
00066 10 CONTINUE
00067 READ(10) (((CEG(I,J,L), I=1,MXO, J=1,MYO, L=1,7)
00068 DO 14 L = 1,MS
00069 DO 14 J = 1,MY
00070 DO 14 I = 1,MX
00071 CAN = CEG(I,J,L)
00072 DO 14 K = 1,MZ
00073 14 CE(I,J,K,L) = CAN
00074 GO TO 18
00075 C ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::
00076 40 CONTINUE
00077 MXYYZ2 = 2*MXYZ
00078 DO 44 N = 1,MXYZ
00079 CE1(N) = 1.
00080 CE1(MXYZ+N) = 0.
00081 44 CE1(MXYZ2+N) = 30.
00082 C44 IF(N.GT.MXY.AND.LT.2*MXY+1) CE1(MXYZ2 + N) = 10.
00083 C ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::
00084 C = = = =
00085 C = = = SCAN FOR MAXIMUM CONCENTRATIONS
00086 C = = = =
00087 18 NAP = -MXYZ + 1
00088 DO 21 N = 1,MS
00089 CMAX = 0.0
00090 NAP = NAP + MXYZ
00091 CMIN = 1000000.
00092 NAPPY = NAP + MXY - 1
00093 DO 20 IJ = NAP,NAPPY
00094 CNAP = CE1(IJ)
00095 CMAX = AMAX1(CMAX,CNAP)
00096 20 CMIN = AMIN1(CMIN,CNAP)
00097 XMAX (N,1) = CMAX
00098 XMIN(N) = CMIN
00099 NAP1 = NAPPY + 1
00100 NAPPY = NAP + MXYZ - 1
00101 DO 22 IJK = NAP1,NAPPY
00102 CNAP = CE1(IJK)
00103 22 CMAX = AMAX1(CMAX,CNAP)
00104 21 XMAX (N,2) = CMAX
00105 RETURN
00106 END

```

```

00001      SUBROUTINE SETUP
00002      COMMON NVERS(6),NDCOD1,NDCOD2,NDCOD3,NPLACE(4),NDATE(3),NHR,NOHRS
00003      1      ,MX,MY,MZ,MS,NS,DX,DY,NSTEP,AVESOL,UF,UFX,DI,LI,LIS,UFY
00004      1      ,INT1,INT,MXY,MXYZ,MXYZS,MXS,MYS,MYXS,MXZ,MYZ,MXZS,MYZS
00005      1      ,DZ(05),CE(37,51,05,18),SE(07,37,51,05)
00006      1      ,IBC(51),BCX(13,51,05,02),BCY(13,37,05,2),SOL(37,51)
00007      1      ,CEG(37,51,07),BCXG(07,51,2),BCYG(07,37,2)
00008      1      ,U(37,51,05,2),D(37,51,05),H2O(37,51),NSPEC(2,07)
00009      1      ,IPRTSP(07)
00010      1      ,UFZ2(05)      ,XMAX(13,2),SL(07,11)
00011      1      ,CC(13),CL(37,51,05,13),XMIN(13),ICOAST(51)
00012      C      DIMENSION U1(1),D1(1),          BGX1(1),BCY1(1),MDATE(3),CE1(1)
00013      C      DIMENSION SE2(MS,MXYZ)
00014      C      DIMENSION SE2(07,1)
00015      EQUIVALENCE (U1,U),(SE2,SE) (BCY1,BCY),(CE1,CE),(BGX1,BGX),(D1,D)
00016      DATA NOTE/1/,ITEST/1/
00017      IF(NOTE.NE.1) GO TO 9
00018      DO 8 IJK = 1,MXYZ
00019      8      SE2(2,IJK) = 0.
00020      9      CONTINUE
00021      IF(NHR.LT.25) GO TO 1
00022      NHR=1
00023      NDATE(2)=NDATE(2)+1
00024      1      CONTINUE
00025      IF(ITEST.GT.0) GO TO 5
00026      READ(5,9992) UONE,UTWO,DONE,BX,BY
00027      9992 FORMAT(5F5.2)
00028      DO 9980 K=1,MZ
00029      DO 9940 IS=1,NS
00030      DO 9930 I=1,MX
00031      BCY(IS,I,K,1)=BY
00032      BCY(IS,I,K,2)=BY
00033      9930 CONTINUE
00034      DO 9940 J=1,MY
00035      BCX(IS,J,K,1)=BX
00036      BCX(IS,J,K,2)=BX
00037      9940 CONTINUE
00038      DO 9980 I=1,MX
00039      DO 9980 J=1,MY
00040      U(I,J,K,1)=UONE
00041      U(I,J,K,2)=UTWO
00042      D(I,J,K)=DONE
00043      SOL(I,J)=0.
00044      H2O(I,J)=0.
00045      9980 CONTINUE
00046      C :::::::::::::::::::::GO TO 6::::::::::::::::::
00047      C :::::::::::::::::::::GO TO 6::::::::::::::::::
00048      C :::::::::::::::::::::MS1 = MS + 1::::::::::::::::::
00049      5      MS1 = MS + 1
00050      IF(NOTE.EQ.1) GO TO 50
00051      DO 10 IU = 10,12
00052      READ(IU) MDATE,MHR
00053      IF(MDATE(1).NE.NDATE(1)) STOP 12
00054      IF(MDATE(2).NE.NDATE(2)) STOP 30
00055      IF(MDATE(3).NE.NDATE(3)) STOP 1675
00056      IF(MHR.NE.NHR) STOP 24

```

```

00057    10   CONTINUE
00058    READ( 10)
00059    50   READ( 10) (((BCXG(L,J,MD,L=1,7),J=1,MY),M=1,2),
00060      1(((BCYG(L,I,MD,L=1,7),I=1,MX),M=1,2)
00061      NOTE = 0
00062      DO 51 J = 1,MY
00063      DO 44 L = 1,MS
00064      CAN1 = BCXG(L,J,1)
00065      CAN2 = BCXG(L,J,2)
00066      DO 44 K = 1,MZ
00067      BCX(L,J,K,1) = CAN1
00068      44   BCX(L,J,K,2) = CAN2
00069      IF(MS.EQ.NS) GO TO 51
00070      DO 45 LL = MS1,NS
00071      DO 45 K = 1,MZ
00072      BCX(LL,J,K,1) = CE( ICOAST(J),J,K,LL)
00073      45   BCX(LL,J,K,2) = CE( IBC(J),J,K,LL)
00074      51   CONTINUE
00075      DO 70 I = 1,MX
00076      DO 60 L = 1,MS
00077      CAN1 = BCYG(L,I,1)
00078      CAN2 = BCYG(L,I,2)
00079      DO 60 K = 1,MZ
00080      BCY(L,I,K,1) = CAN1
00081      60   BCY(L,I,K,2) = CAN2
00082      IF(MS.EQ.NS) GO TO 70
00083      DO 65 LL = MS1,NS
00084      DO 65 K = 1,MZ
00085      BCY(LL,I,K,1) = CE( I,1,K,LL)
00086      65   BCY(LL,I,K,2) = CE( I,MY,K,LL)
00087      70   CONTINUE
00088      READ(11) ((SOL(I,J),I=1,MX),J=1,MY),((H2O(I,J),I=1,MX),J=1,MY)
00089      READ(11) (((D(I,J,K),I=1,MX),J=1,MY),K=1,MZ)
00090      READ(11) (((U(I,J,K,1),I=1,MX),J=1,MY),K=1,MZ),
00091      1(((U(I,J,K,2),I=1,MX),J=1,MY),K=1,MZ)
00092      READ(12) (((SE(1,I,J,K),I=1,MX),J=1,MY),K=1,MZ),
00093      1(((SE(L,I,J,K),L=3,MS),I=1,MX),J=1,MY),K=1,MZ)
00094      GO TO 81
00095      C ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::
00096      6   READ(5,9991) IS1,JS1,KS1,IS2,JS2,KS2
00097      C ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::
00098      9991 FORMAT(40I2)
00099      DO 9950 I=1,MX
00100      DO 9950 J=1,MY
00101      DO 9950 K=1,MZ
00102      SE(1,I,J,K)=0.
00103      IF(I.EQ.IS1.AND.J.EQ.JS1.AND.K.EQ.KS1) SE(1,I,J,K)=1.
00104      SE(2,I,J,K)=0.
00105      IF(I.EQ.IS2.AND.J.EQ.JS2.AND.K.EQ.KS2) SE(2,I,J,K)=1.
00106      C ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::::::::::: ::::::::::::
00107      9950 CONTINUE
00108      81   M=0
00109      DO 20 K=1,MZ
00110      UFZ2K=UFZ2(K) *.75
00111      DO 20 IJ=1,MAXY
00112      M=M+1

```

```

00001      SUBROUTINE TRACE
00002      COMMON NVERS(6),NDCOD1,NDCOD2,NDCOD3,NPLACE(4),NDATE(3),NHR,NOHRS
00003      ,MX,MY,MZ,MS,NS,DX,DY,NSTEP,AVESOL,UF,UFX,DI,LI,LIS,UFY
00004      ,INT1,INT,MXY,MXYZ,MXYZS,MXS,MYS,MXYS,MXZ,MYZ,MXZS,MYZS
00005      ,DZ(05),CE(37,51,05,13),SE(07,37,51,05)
00006      ,IBC(51),BCX(13,51,05,02),BCY(13,37,05,2),SOL(37,51)
00007      ,CEG(37,51,07),BCNG(07,51,2),BCYG(07,37,2)
00008      ,U(37,51,05,2),DC(37,51,05),E20(37,51),NSPEC(2,07)
00009      ,IPRTSP( 07)
00010      ,UFZ2(05) ,XMAX(13,2),SL( 07,11)
00011      ,CC( 13),CL(37,51,05,13),XMIN(13),ICOAST(51)
00012      DIMENSION SL1(1),SE1(1),CL1(1),CE1(1),BCX1(1),BCY1(1)
00013      EQUIVALENCE (SL1,SL),(SE1,SE),(CL1,CL),(CE1,CE),(BCX1,BCX)
00014      ,(BCY1,BCY)
00015      DO 390 K=1,MZ
00016      Z0 = DZ(K)
00017      SUMZ = Z0
00018      IF(K.EQ.1) GO TO 301
00019      ZM = DZ(K-1)
00020      SUMZ = Z0 + ZM
00021      IF(K.EQ.MZ) GO TO 302
00022      ZP = DZ(K+1)
00023      SUMZ = SUMZ + ZP
00024      ZU = ZP
00025      GO TO 303
00026      ZU = ZM
00027      301   KAP = (K - 1)*MXY
00028      KAPS = (K - 1)*MXYS
00029      DO 390 JJ12=1,MY
00030      IBCJ = IBC(JJ12)
00031      ICOASJ = ICOAST(JJ12)
00032      DO 390 II12=ICOASJ,IBCJ
00033      I=II12
00034      J=JJ12
00035      X = I + .5
00036      Y = J + .5
00037      LM = (I - 1)*MS + (J - 1)*MXS+ KAPS
00038      AVESOL=SOL(I,J)
00039      AVEDIF=D(I,J,K)
00040      DO 310 L = 1,MS
00041      310   SL1(LI+L) = SE1(LM+L)
00042      LN = LI
00043      DO 320 M = 1,INT1
00044      Y = Y - U(I,J,K,2)
00045      X = X - U(I,J,K,1)
00046      I = X
00047      J = Y
00048      IF(J.LT.1) GO TO 321
00049      IF(J.GT.MY) GO TO 322
00050      IF(I.LT.ICOAST(J)) GO TO 323
00051      IF(I.GT.IBC(J)) GO TO 324
00052      LN = LN - MS
00053      LM = (I - 1)*MS + (J - 1)*MXS+ KAPS
00054      AVESOL=AVESOL+SOL(I,J)
00055      AVEDIF=AVEDIF+D(I,J,K)
00056      DO 330 L = 1,MS

```

```

00057 330    SL1(LN+L) = SE1(LM+L)
00058 320    CONTINUE
00059      EX = 0.
00060      ISTART = 1
00061      AVESOL = AVESOL/INT
00062      DD = AVEDIF / INT
00063      IJK = I + (J-1)*MX + KAP
00064      LL = IJK - MXXYZ
00065      IF(K.EQ.1) GO TO 349
00066      IF(K.EQ.MZ) GO TO 351
00067      LLL = LL + MXY
00068      LLLL = LL - MXY
00069      IF(DD.LT.5.) EX = EXP(-3.*DD)
00070      DO 325 L = 1 , NS
00071      LL = LL + MXXYZ
00072      LLL = LLL + MXXYZ
00073      LLLL = LLLL + MXXYZ
00074      CAP = (CE1(LL)*Z0 + CE1(LLL)*ZP + CE1(LLLL)*ZMD)/SUMZ
00075 325      CC(L) = CAP + (CE1(LL) - CAP)*EX
00076      GO TO 366
00077 349      LLL = LL + MXY
00078      GO TO 360
00079 351      LLL = LL - MXY
00080 360      IF(DD.LT.5.) EX = EXP(-2.*DD)
00081      DO 365 L = 1,NS
00082      LL = LL + MXXYZ
00083      LLL = LLL + MXXYZ
00084      CAP = (CE1(LL)*Z0 + CE1(LLL)*ZU)/SUMZ
00085 365      CC(L) = CAP + (CE1(LL) - CAP)*EX
00086 366      IF((I-II12)**2+(J-JJ12)**2.GT. 4.) GO TO 380
00087      JA=Y+.5
00088      IF(JA.GT.MY) JA=MY
00089      JA1=JA-1
00090      IF(JA1.LT.1) JA1=1
00091      IA=X+.5
00092      IF( IA.GT.MX) IA=MX
00093      IA1=IA-1
00094      IF( IA1.LT.1) IA1=1
00095      FX=X-(IA1+.5)
00096      FX1=1.-FX
00097      FY=Y-(JA1+.5)
00098      FY1=1.-FY
00099      K1M=(K-1)*MXY-MXYZ
00100      L11=IA+(JA1)*MX+K1M
00101      L12=IA+(JA1-1)*MX+K1M
00102      L21=IA1+(JA1)*MX+K1M
00103      L22=IA1+(JA1-1)*MX+K1M
00104      LL = IJK - MXXYZ
00105      DO 388 L=1,NS
00106      L11=L11+MXXYZ
00107      L12=L12+MXXYZ
00108      L21=L21+MXXYZ
00109      L22=L22+MXXYZ
00110      LL = LL + MXXYZ
00111      CC(L)=FX*FY*CE1(L11)+FX*FY1*CE1(L12)+FX1*FY*CE1(L21)
00112      1 +FX1*FY1*CE1(L22) + CC(L) - CE1(LL)

```

```

00113   388 CC(L) = AMAX1(CC(L),0.)
00114   380 CONTINUE
00115   C CALL INERT(ISTART)
00116   CALL CHEMC(ISTART,I,J)
00117   C
00118   C IF(NSTEP.GT.1)CALL CHEMC(ISTART)
00119   C:::::::::::ML=II12+(JJ12-1)*MX+KAP-MXYZ
00120   ML=II12+(JJ12-1)*MX+KAP-MXYZ
00121   DO 350 L = 1,NS
00122   ML=ML+MXYZ
00123   350 CL1(ML) = CC(L)
00124   GO TO 390
00125   321 IF(I.LT.ICOAST(1)) I = ICOAST(1)
00126   IF(I.GT.IBC(1)) I = IBC(1)
00127   J=1
00128   LL = (I - 1 + (K-1)*MX)*NS
00129   DO 355 L = 1,NS
00130   355 CC(L) = BCY1(LL + L)
00131   GO TO 379
00132   322 IF(I.LT.ICOAST(MY)) I = ICOAST(MY)
00133   IF(I.GT.IBC(MY)) I = IBC(MY)
00134   J=MY
00135   LL = (I - 1 + (K-1+MZ)*MX)*NS
00136   DO 356 L = 1,NS
00137   356 CC(L) = BCY1(LL + L)
00138   GO TO 379
00139   323 IF(J.LT.1) J = 1
00140   IF(J.GT.MY) J = MY
00141   I=ICOAST(J)
00142   LL = (J - 1 + (K-1)*MY)*NS
00143   DO 357 L = 1,NS
00144   357 CC(L) = BCX1(L + LL)
00145   GO TO 379
00146   324 IF(J.LT.1) J = 1
00147   IF(J.GT.MY) J = MY
00148   I=IBC(J)
00149   LL = (J - 1 +(K-1+MZ)*MY)*NS
00150   DO 358 L = 1,NS
00151   358 CC(L) = BCX1(L + LL)
00152   379 ISTART = INT - M
00153   MMS = ISTART*MS
00154   DO 311 L = 1,MMS
00155   311 SL1(L) = 0.
00156   AVESOL = AVESOL/M
00157   GO TO 380
00158   390 CONTINUE
00159   DO 411 I = 1,MX
00160   DO 411 J = 1,MY
00161   DO 411 L = 1,NS
00162   411 CALL DIFUS(CL(I,J,1,L),D(I,J,1),DZ,MZ,MXY)
00163   DO 400 J = 1,MY
00164   JX = (J - 1)*MX
00165   ICOASJ = ICOAST(J)
00166   IBCJ = IBC(J)
00167   DO 400 L = 1,NS
00168   LX = (L - 1)*MXYZ + JX

```

```

00169      DO 400 K = 1,MZ
00170      JKL = LX + (K - 1)*MXY
00171      BD = CL1(JKL + ICOASJ)
00172      IB = JKL + 1
00173      IE = JKL + ICOASJ
00174      DO 402 I = IB,IE
00175      402 CL1(I) = BD
00176      BD = CL1(JKL+IBCJ)
00177      IB = JKL + IBCJ
00178      IE = JKL + MX
00179      DO 404 I = IB,IE
00180      404 CL1(I) = BD
00181      400 CONTINUE
00182      NAP = -MXYZ + 1
00183      DO 398 N = 1,NS
00184      CMAX = 0.0
00185      NAP = NAP + MXYZ
00186      CMIN = 1000000.
00187      NAPPY = NAP + MXY - 1
00188      DO 396 IJ = NAP,NAPPY
00189      CNAP = CL1(IJ)
00190      CMAX = AMAX1(CMAX,CNAP)
00191      CMIN = AMIN1(CMIN,CNAP)
00192      396 CE1(IJ) = CNAP
00193      XMAX(N,1) = CMAX
00194      XMIN(N) = CMIN
00195      NAP1 = NAPPY + 1
00196      NAPPY = NAP + MXYZ -1
00197      DO 397 IJK = NAP1,NAPPY
00198      CNAP = CL1(IJK)
00199      CMAX = AMAX1(CMAX,CNAP)
00200      397 CE1(IJK) = CNAP
00201      398 XMAX(N,2) = CMAX
00202      RETURN
00203      END

```

SUBROUTINE OUTPUT(MYDSK, NEDIT)

..... OUTPUT IS CALLED INITIALLY AND EVERY SUBSEQUENT HOUR TO  
PRINT THE EULERIAN CONCENTRATIONS CE(I,J,K,L) ,313,  
, I2)

COMMON /VERS(6), NDCOD1, NDCOD2, NDCOD3, NPLACE(4), NDATE(3), NHR, NOHRS  
, MX, MY, MZ, MS, NS, DX, DY, NSTEP, AVESOL, UF, UFX, DI, LI, LIS, UFY  
, INT1, INT, MXY, MXYZ, MXYS, MXS, MYS, MXS, MYZ, MXZ, MYZ, MXZS, MYZS  
, DZ(05), CE(37,51,05,13), SE(07,37,51,05) ,T90 ,  
, IBC(51), BCX(13,51,05,02), BCY(13,37,05,2), SOL(37,51) (1), NDATE(2)  
, CEG(37,51,07), BCXG(07,51,2), BCYG(07,37,2)  
, U(37,51,05,2), D(37,51,05), H2O(37,51), NSPEC(2,07)  
, IPRTSP( 07)  
, UFZ2(05) , XMAX(13,2), SL( 07,11) .....

1 DIMENSION ICE(100), ISTART(4)  
DATA NOTE/1/, IBL/1000/

IF(NOTE.EQ.0) GO TO 1  
ISTART(1) = NHR  
ISTART(2) = NDATE(1)  
ISTART(3) = NDATE(2)  
ISTART(4) = NDATE(3)  
NOTE = 0

.... WRITE THE TAPE .....

1 CONTINUE ,T90 ,  
IF(MYDSK.EQ.1) WRITE(13,101) NPLACE, NVERS, NDCOD1,  
1 NDCOD2, NDCOD3, ISTART  
IF(MYDSK.EQ.1) WRITE(13,100) NHR, NDATE  
FORMAT' THE FOLLOWING OUTPUT IS FOR ', I2, '00PST', 3(I2, '...') ATE  
00 IF(MYDSK.EQ.1) WRITE(13) (((CE(I,J,K,L), I=1,MX), J=1,MY),  
1 K=1, MZ), L=1, NS) .....

.... LOOP OVER SPECIES .....

1 LU=6 IE ' ,2X,  
DO 40 N = 1, MS  
IF(N.EQ.2.OR.N.EQ.4.AND.NEDIT.EQ.1) LU=8  
1, MS)  
2 IF(IPRTSP( N).NE.1) GO TO 40

.... FIND FACTOR FOR A TWO DIGIT PRINT .....

F = 1.  
FACTOR = 1.  
IF(XMAX (N,2).LT.1.0E-20) GO TO 11  
LOGX = IF IX ALOG10(XMAX (N,2)))  
IF(XMAX (N,2).LT.1.0) LOGX = LOGX - 1  
FACTOR = 10.\*\*(LOGX - 1)  
F = 1./FACTOR  
CONTINUE

.... LOOP OVER LEVELS AND WRITE HEADERS .....

```

00001      SUBROUTINE INERT(ISTART)
00002      COMMON NVERS(6),NDCOD1,NDCOD2,NDCOD3,NPLACE(4),NDATE(3),NHR,NOHRS
00003      1      ,MX,MY,MZ,MS,NS,DX,DY,NSTEP,AVESOL,UF,UFX,DI,L1,LIS,UFY
00004      1      ,INT1,INT,MXY,MXYZ,MXYZS,MXS,MYS,MXYS,MXZ,MYZ,MXZS,MYZS
00005      1      ,DZ(05),CE(37,51,05,13),SE(07,37,51,05)
00006      1      ,IBC(51),BCX(13,51,05,02),BCY(13,37,05,2),SOL(37,51)
00007      1      ,CEG(37,51,07),BCXG(07,51,2),BCYG(07,37,2)
00008      1      ,U(37,51,05,2),D(37,51,05),H20(37,51),NSPEC(2,07)
00009      1      ,IPRTSP( 07)
00010      1      ,UFZ2(05)      ,XMAX(13,2),SL( 07,11)
00011      1      ,CC( 13),CL(37,51,05,13),XMIN(13),ICOAST(51)
00012      C
00013      C
00014      ISTAR1 = ISTART + 1
00015      DO 10 L = 1,MS
00016      SUM = (SL(L,ISTART) + SL(L,INT))/2.
00017      IF(ISTAR1.GT.INT1) GO TO 10
00018      DO 5 N = ISTAR1,INT1
00019      5      SUM = SUM + SL(L,N)
00020      10     CC(L) = CC(L) + SUM*DI
00021      RETURN
00022      END

```

```

00001      SUBROUTINE DIFUS(C,D,DZ,N,ISKP)
00002      DIMENSION C(10),D(10),DZ(10),DIP1(10),DIM1(10),A(10),B(10)
00003      NM1=N-1
00004      IP=1-ISKP
00005      DO 10 I=1,NM1
00006      IP=IP+ISKP
00007      IP1=IP1+ISKP
00008      D2=   (D(IP)*DZ(I)    +D(IP1)*DZ(I+1)) /((DZ(I)+DZ(I+1))**2)
00009      D2=.3333333*D2
00010      DIP1(I)=D2*DZ(I+1)
00011      10  DIM1(I+1)=D2*DZ(I)
00012      A(I)=DIP1(1)/(1.+DIP1(1))
00013      C  B(I)=((1.-DIP1(1))*C(1)+DIP1(1)*C(1+ISKP))/(1.+DIP1(1))
00014      B(I)=C(I)/(1.+DIP1(1))
00015      IP=1-ISKP
00016      DO 20 I=2,NM1
00017      IP=IP+ISKP
00018      IP1=IP1+ISKP
00019      IP2=IP1+ISKP
00020      BTM=1.+DIP1(I)+DIM1(I)*(1.-A(I-1))
00021      A(I)=DIP1(I)/BTM
00022      C  20 B(I)=(DIM1(I)*(B(I-1)+C(IP)))
00023      C  1 +(1.-DIM1(I)-DIP1(I))*C(IP1)+DIP1(I)*C(IP2))/BTM
00024      20 B(I)=(C(IP1)+DIM1(I)*B(I-1))/BTM
00025      BTM=1.+DIM1(N)*(1.-A(NM1))
00026      C  C(IP2)=(DIM1(N)*(B(NM1)+C(IP1))+(1.-DIM1(N))*C(IP2))/BTM
00027      C(IP2)=(C(IP2)+B(NM1)*DIM1(N))/BTM
00028      DO 30 J=1,NM1
00029      I=N-J
00030      C(IP1)=A(I)*C(IP2)+B(I)
00031      IP2=IP2-ISKP
00032      IP1=IP1-ISKP
00033      30 CONTINUE
00034      RETURN
00035      END

```

```

00001      SUBROUTINE CHEMC( INTI, II, JJ)
00002      C ****
00003      C *          C H E M C   SOLVES AN UPDATED VERSION OF THE LIRAQ-2      *
00004      C *          PHOTOCHEMICAL MODEL ( R. J. GELINAS, P. D.      *
00005      C *          SKEWS-CGX, 'TROPOSPHERIC PHOTOCHEMICAL      *
00006      C *          MECHANISMS', AMERICAN CHEMICAL SOCIETY      *
00007      C *          173RD NATIONAL MEETING, MARCH 1977 ).      *
00008      C *          THE MODEL EMPOLYES THREE HYDROCARBON      *
00009      C *          SPECIES AND A PRELIMINARY SULFER DIOXIDE      *
00010      C *          REACTION SET IN A 56 STEP LUMPED PARAMETER      *
00011      C *          KINETIC MECHANISM. FINAL CONCENTRATIONS      *
00012      C *          ARE OBTAINED VIA PREDICTOR-CORRECTOR      *
00013      C *          EXPONENTIAL SOLUTION TO THE SET OF REACTIONS.      *
00014      C *
00015      C ****
00016      C
00017      COMMON NVERS(6),NDCOD1,NDCOD2,NDCOD3,NPLACE(4),NDATE(3),NHR,NOHRS
00018      1     ,MX,MY,MZ,MS,NS,DY,NSTEP,AVESOL,UF,UFXY,DI,LI,LIS,UFY
00019      1     ,INT1,INT,MXY,MXYZ,MXYZS,MXS,MYS,MXYS,MXZ,MYZ,MXZS,MYZS
00020      1     ,DZ(05),CE(37,51,05,13),SE(07,37,51,05)
00021      1     ,IBC(51),BCX(13,51,05,02),BCY(13,37,05,2),SOL(37,51)
00022      1     ,CEG(37,51,07),BCXG(07,51,2),BCYG(07,37,2)
00023      1     ,U(37,51,05,2),D(37,51,05),H2O(37,51),NSPEC(2,07)
00024      1     ,IPRTSP( 07)
00025      1     ,UFZ2(05)           ,XMAX(13,2),SLC( 07,11)
00026      1     ,CC( 13),CL(37,51,05,13),XMIN(13),ICOAST(51)
00027      DIMENSION CON(15),CON1(15),SR(15),SR1(15),AK(58),BK(58),A(15)
00028      ,B(15),K(58),SK(15),CON2(15),IC(15),CONN(15) ,KTEMP(58)
00029      REAL NO,NO2,NO3,N205,K,NOX,K4 ,KTEMP
00030      DATA SR,SR1,SK/45#0./
00031      DATA ALPHA,O2,ZM,AVG,STP/0.5.,22E06,1.E06,6.0225E23,22.4E09/
00032      DATA ERR,ERR1,ERR2/0.005,0.001,0.01/
00033      DATA DTI/1.E-6/
00034      DATA ALPHA1/0.5/
00035      DATA TFAC/0.5/
00036      DATA AK/1.2E-02,7.2E-04,5.5E-06,2.8E-03,5.1E-04,2.2E-05,1.9E-03,
00037      .    2.7E-05,2.7E-05,1.0E-34,9.0E-13,4.0E-33,9.1E-12,3.5E-32,
00038      .    1.1E-13,3.8E-12,5.7E-14,8.7E-12,3.0E-16,7.0E-12,7.0E-12,
00039      .    2.0E-13,3.0E-14,1.3E-13,3.0E-12,2.0E-11,1.4E-13,1.0E-11,
00040      .    7.0E-15,7.0E-15,5.0E-11,3.0E-11,4.1E-11,4.0E-11,4.0E-12,
00041      .    3.0E-11,8.0E-13,3.0E-15,3.0E-11,3.0E-15,8.0E-13,0.0
00042      .    5.0E-12,6.5E-12,3.0E-14,5.0E-14,5.0E-14,1.6E-13,6.7E-14,
00043      .    6.7E-14,2.0E-15,4.0E-13,9.0E-16,2.4E-32,3.4E-32,1.0E-15,
00044      .    4.9E-7,1.9E-5 /
00045      DATA BK/0.,0.,0.,0.,0.,0.,0.,-510.,1200.,-940.,0.,-300.,
00046      .    2450.,0.,10600.,0.,3300.,0.,0.,0.,0.,0.,0.,0.,360.,
00047      .    1900.,1900.,350.,350.,2000.,900.,900.,350.,3700.,900.,
00048      .    350.,650.,3750.,0.,300.,600.,0.,0.,0.,3300.,0.,0.,0.,
00049      .    0.,0.,0.,1130.,0.,0.0,0.0/
00050      DATA TEMP/298.0/
00051      DATA SO2/0.0/
00052      DATA IBUG/0/
00053      DATA ISKIP/0/,CNT/0.9E-6/
00054      DATA TIME/60./
00055      DATA IC/13,3,1,2,4,5,6,7,8,9,10,11,12,14,15/
00056      C

```

```

00057 C-----COMPUTE RATE CONSTANTS
00058 CF = 22.4E09*TEMP/6.E23/273.
00059 H2O = H2O(II,JJ)
00060 IF( ISKIP. NE.0) GO TO 200
00061 DO 100 I=1,58
00062 K(I)=AK(I)
00063 IF( BK(I).NE.0.0) K(I)=AK(I)*EXP(-BK(I)/TEMP)
00064 100 CONTINUE
00065 DO 110 I=1,58
00066 K(I)=K(I)*60.0
00067 IF( I.LT.10) GO TO 110
00068 IF( I.EQ.17.OR.I.EQ.57.OR.I.EQ.58) GO TO 110
00069 K(I)=K(I)/CF
00070 IF( I.EQ.10.OR.I.EQ.12.OR.I.EQ.14.OR.I.EQ.54.OR.I.EQ.55)
00071 K(I)= K(I)*ZM/CF
00072 110 KTEMP(I) = K(I)
00073 ISKIP=100
00074 C-----SET PHOTOCHEMICAL RATE CONSTANTS
00075 200 CONTINUE
00076 DO 210 I=1,9
00077 K(I)= KTEMP(I)*AVESOL/ 32.
00078 210 CONTINUE
00079 K(57)= KTEMP(57)*AVESOL/ 32.
00080 K(58)= KTEMP(58)*AVESOL/ 32.
00081 K(6)= K(6)*H2O *6.0/ ZM
00082 IAM = 1
00083 IF( IBUG.EQ.1) WRITE(6,900) IAM,K
00084 C-----SET TIME
00085 TOLT = 0.0
00086 DT = DTI
00087 DTS = TIME / FLOAT(INT1)
00088 DELTM = DTS*FLOAT( INT - INT1)
00089 C-----INITILIZE CONCENTRATIONS AND SOURCE TERMS
00090 DO 220 I=1,7
00091 J=IC(I)
00092 SR(J) = SL(I,INT1)
00093 220 CONTINUE
00094 C INITIAL ASSIGNMENT
00095 DO 230 I=1,13
00096 J=IC(I)
00097 230 CON(J)=CC(I)
00098 C
00099 CON(14)=1.E-10
00100 CON(15)=1.E-10
00101 300 CONTINUE
00102 C
00103 C-----ASSIGN SOURCE TERM
00104 TNEW = TOLT + DT
00105 TST = TNEW/DTS
00106 INTX = INT1 + IFIX(TST)
00107 INTX = MIN0(INTX,INT1)
00108 DO 310 I=1,7
00109 J=IC(I)
00110 SR1(J) =(SL(I,INTX+1) - SL(I,INTX)) * (TST-IFIX(TST))
00111 +SL(I,INTX)
00112 SK(J) =(SR(J) + SR1(J)) * 0.5

```

```

00113      310 CONTINUE
00114      C
00115      C      DEFINE POLLUTANTS
00116      C      ASSIGN 500 TO KJMP
00117      NO2    = CON(1)
00118      NO     = CON(2)
00119      O3     = CON(3)
00120      HC1    = CON(4)
00121      HC2    = CON(5)
00122      HC3    = CON(6)
00123      N205   = CON(7)
00124      HN03   = CON(8)
00125      HN02   = CON(9)
00126      HO2    = CON(10)
00127      RN02   = CON(11)
00128      H202   = CON(12)
00129      CO     = CON(13)
00130      R02=CON(14)
00131      RC03=CON(15)
00132      C
00133      400 CONTINUE
00134      IAM = 2
00135      IF( IBUG.EQ.1) WRITE(6,900) IAM,NO2,NO,O3,HC1,HC2,HC3,N205,HN03,
00136      .      HN02,HO2,RN02,H202,CO,R02,RC03
00137      C
00138      C ****
00139      C * STEADY STATE SPECIES *
00140      C ****
00141      RO=(K(7)*RN02+K(29)*O3*HC1+R02*(K(43)*NO+2.*K(49)*R02+K(50)*HO2
00142      .      +K(51)*S02))/(K(46)*NO2+K(47)*NO+K(48)*O2+K(52)*S02)
00143      IF(AVESOL.NE.0.) GO TO 410
00144      O=0.
00145      411 C2=K(20)*NO2+K(21)*NO+K(24)*HN03+K(26)*HO2+K(27)*CO
00146      .      +(K(31)+K(32))*HC1+K(34)*HC2+(K(36)+K(39))*HC3+K(54)*S02
00147      B2=(K(2)*HN02+2.*K(3)*H202+K(6)*O3)+HO2*(K(22)*NO+K(50)*R02+K(53)
00148      .      *S02)+K(57)*HN03)
00149      HO=0.
00150      IF(C2.NE.0.) HO=B2/C2
00151      C3=K(4)+K(16)*NO2+K(18)*NO+(K(38)+K(40))*HC3
00152      D3=K(15)*O3*NO2+K(17)*N205
00153      E3=K(24)*HN03
00154      NO3=1.E-10
00155      IF(C3.NE.0.) NO3=(D3+HO*B3)/C3
00156      GO TO 490
00157      410 IF(K(4).NE.0.) GO TO 420
00158      A1=(K(10)*O2+K(12)*NO+(K(13)+K(14))*NO2+K(28)*HC1+K(33)*HC2
00159      .      +K(35)*HC3+K(55)*S02)
00160      B1=-(K(1)*NO2+K(5)*O3+K(58)*N205)
00161      O=1.E-10
00162      IF(A1.NE.0.) O=B1/A1
00163      GO TO 411
00164      420 CONTINUE
00165      A1=(K(10)*O2+K(12)*NO+(K(13)+K(14))*NO2+K(28)*HC1+K(33)*HC2
00166      .      +K(35)*HC3+K(55)*S02)/K(4)
00167      B1=-(K(1)*NO2+K(5)*O3+K(58)*N205)/K(4)
00168      C2=K(20)*NO2+K(21)*NO+K(24)*HN03+K(26)*HO2+K(27)*CO

```

```

00169      +(K(31)+K(32))*HC1+K(34)*HC2+(K(36)+K(39))*HC3+K(54)*S02
00170      A2=(K(33)*HC2+K(35)*HC3)/C2
00171      B2=(K(2)*HN02+2.*K(3)*H202+K(6)*03)+H02*(K(22)*NO+K(50)*R02+K(53)
00172      *S02)+K(57)*HN03)/C2
00173      A3=K(14)*N02
00174      B3=K(24)*HN03
00175      C3=K(4)+K(16)*N02+K(18)*NO+(K(38)+K(40))*HC3
00176      D3=K(15)*03*N02+K(17)*N205
00177      O=(D3+B2*B3-B1*C3)/(A1*C3-A2*B3-A3)
00178      HO=A2*O+B2
00179      NO3=(D3+HO*B3+O*A3)/C3
00180      490 CONTINUE
00181      IF(O.LT.0.) O=-0
00182      IF(NO3.LT.0.) NO3=-NO3
00183      IF(HO.LT.0.) HO=-HO
00184      IF(R0.LT.0.) R0=-R0
00185      C
00186      IAM = 3
00187      IF( IBUG.EQ.1) WRITE(6,900) IAM,R02,HO,R0,RC03,NO3,O
00188      IF( IBUG.EQ.1) WRITE(6,900) IAM,A1,B1,A2,B2,C2,A3,B3,C3,D3
00189      C
00190      ****
00191      C * BASIC SPECIES *
00192      ****
00193      A(1) = K(4)*NO3 + K(17)*N205 + NO *( K(11)*03 + K(12)*0 +
00194      . K(18)*2.0*NO3 + K(22)*H02 + K(43)*R02 +
00195      . K(44)*RC03 ) + SK(1)
00196      B(1) = K(1) + O *( K(13)+K(14) ) + K(15)*03 + K(16)*NO3 +
00197      . K(20)*HO + K(23)*H02 + K(45)*RC03 + K(46)*R0
00198      C
00199      A(2) = ( K(1)+K(13)*0 ) * N02 + K(2)*HN02 + K(7)*RN02 + SK(2)
00200      B(2) = K(12)*0 + K(18)*NO3 + K(21)*HO + K(11)*03 + K(22)*H02 +
00201      . (K(42)+K(43))*R02 + K(44)*RC03 + K(47)*R0
00202      C
00203      A(3) = K(10)*0*02 + SK(3)
00204      B(3) = K(5) + K(6) + K(11)*NO + K(15)*N02 + (K(29)+K(30))*HC1
00205      C
00206      A(4) = SK(4)
00207      B(4) = K(28)*0 + (K(29)+K(30))*03 + (K(31)+K(32))*HO
00208      C
00209      A(5) = SK(5)
00210      B(5) = K(33)*0 + K(34)*HO
00211      C
00212      A(6) = SK(6) + HC1 *( K(29)*03 + K(31)*HO ) + K(48)*R0*02
00213      B(6) = K(8) + K(9) + K(35)*0 + (K(36)+K(39))*HO + K(37)*H02 +
00214      . K(41)*R02 + (K(38)+K(40))*NO3
00215      A(13) = SK(13)
00216      B(13) = 0.0
00217      C
00218      A(14) = SK(14)
00219      B(14) = K(51)*R02 + K(52)*R0 + K(53)*H02 + K(54)*HO +
00220      . K(55)*0 + K(56)*RC03
00221      C
00222      ****
00223      C * MINOR SPECIES *
00224      ****

```

```

00225      A(7) = K(16)*N03*N02 + SK(7)
00226      B(7) = K(17) + K(19)*H20+K(58)
00227      C
00228      A(8) = 2.0*K(19)*H20*N205 + K(20)*H0*N02 + (K(38)+K(40)) *
00229          (HC3 * N03 ) + SK(8)
00230      C
00231      B(8) = K(24)* H0+K(57)
00232      C
00233      A(9) = K(21)*H0*N0 + K(23)*H02*N02 + SK(9)
00234      C
00235      A(10) = HC3 * ( 2.0*K(8) + K(9) + K(35)*0 + K(39)*H0 + K(40)*N03 )
00236          + HC1 * ( ALPHA1*K(28)*0 + K(29)*03 ) + K(27)*C0*H0
00237          + K(48)*R0*02 + K(52)*S02*R0 + K(54)*S02*H0 + SK(10)
00238      C
00239      B(10) = K(22)*N0 + K(23)*N02 + 2.*K(25)*H02 + K(26)*H0 + K(37)*HC3
00240          + K(50)*H02 + K(53)*S02
00241      C
00242      A(11) = K(47)*R0*N0 + SK(11)
00243      C
00244      B(11) = K(7)
00245      C
00246      A(12) = K(25)*H02*H02 + K(37)*HC3*H02 + SK(12)
00247      C
00248      B(12) = K(3)
00249      C
00250      A(14)=K(9)*HC3+K(28)*0*HC1+HC1*H0*(K(31)+K(32))+K(33)*0*HC2
00251          +K(34)*H0*HC2+RC03*( N0*K(44)+S02*K(56))+SK(14)
00252      C
00253      B(14)=K(41)*HC3+N0*(K(42)+K(43))+2.*R02*K(49)+K(50)*H02+K(51)*S02
00254      C
00255      IAM = 4
00256      IF( IBUG.EQ.1) WRITE(6,900) IAM,A,B
00257      900 FORMAT(//1X,' IAM = ',I2/(1X,1P15E08.1))
00258      GO TO KJMP, (500,600)
00259      C
00260      PASS ONE
00261      500 DO 521 I=1,15
00262          T = B(I)*DT
00263          T2= T*.5
00264          DT05=DT*.5
00265          IF( T.GT.1.0E-06 ) GO TO 510
00266          CON1(I) = A(I)*DT + CON(1)
00267          CONN(I)=A(I)*DT05+CON(1)
00268          GO TO 520
00269          510 IF( T.GT.75. ) GO TO 515
00270          AOB=A(I)/B(I)
00271          EXPT=EXP(-T2)
00272          CMAOBE=(CON(I)-AOB)*EXPT
00273          CONN(I)=AOB+CMAOBE
00274          CON1(I)=AOB+CMAOBE*EXPT
00275          GO TO 520
00276          515 CON1(I) = A(I)/B(I)
00277          CONN(I)=CON1(I)
00278          520 CON1(I) = AMAX1(0.0,CON1(I))
00279          521 CONN(I)=AMAX1(CONN(I),0.0)
00280      C
00281      N02      = CONN(1)

```

```

00281      NO      = CONN(2)
00282      O3      = CONN(3)
00283      HC1     = CONN(4)
00284      HC2     = CONN(5)
00285      HC3     = CONN(6)
00286      N205    = CONN(7)
00287      HN03    = CONN(8)
00288      HN02    = CONN(9)
00289      H02     = CONN(10)
00290      RN02    = CONN(11)
00291      H202    = CONN(12)
00292      CO      = CONN(13)
00293      R02=CONN(14)
00294      RCO3=CONN(15)
00295      IAM = 5
00296      IF( IBUG.EQ.1) WRITE(6,900) IAM,CON,CON1,CONN
00297      C
00298      ASSIGN 600 TO KJMP
00299      GO TO 400
00300      600 TEST= 0.0
00301      C
00302      PASS TWO
00303      DO 630 I=1,15
00304      T = B(I)*DT
00305      IF( T.GT.1.E-06) GO TO 610
00306      CON2(I) = A(I)*DT + CON(I)
00307      GO TO 620
00308      610 IF( T.GT. 75.) GO TO 615
00309      CON2(I) = A(I)/B(I) +(CON(I) - A(I)/B(I)) * EXP(-T)
00310      GO TO 620
00311      615 CON2(I) = A(I)/B(I)
00312      620 CON2(I) = AMAX1(0.0,CON2(I))
00313      C-----ERROR TEST
00314      IF( CON1(I)+CON2(I).LT.1.E-04) GO TO 630
00315      TEST = AMAX1(TEST,ABS((CON2(I)-CON1(I))/(CON2(I)+CON1(I))))
00316      630 CONTINUE
00317      IAM = 6
00318      IF( IBUG.EQ.1) WRITE(6,900) IAM,CON2,TEST,ERR2
00319      IF( TEST.LT.ERR2) GO TO 640
00320      C-----DECREASE TIME STEP
00321      DT = DT * TFAC
00322      GO TO 300
00323      C-----UPDATE TIME
00324      640 CONTINUE
00325      TOLT= TNEW
00326      IAM = 10
00327      IF( TOLT.LT.CNT) GO TO 699
00328      CNT = CNT + 6.
00329      IF( IBUG.EQ.2) WRITE(6,900) IAM,CON2,SK,TOLT,DT
00330      699 CONTINUE
00331      IF( TOLT.GE.DELTM) GO TO 700
00332      TEST =AMAX1(TEST,ERR1)
00333      DT = AMIN1(DT*ERR/TEST, DELTM-TOLT + 1.E-04,TIME*0.1)
00334      DO 650 I=1,15
00335      CON(I) = CON2(I)
00336      SR(I) = SR1(I)
00337      650 CONTINUE

```

```

00001      SUBROUTINE PL(Z,NX,NY,ZMAX,ZMIN)
00002      C*****
00003      C*
00004      C*          P L  IS A PRINTER-PLOTTER ROUTINE WHICH      *
00005      C*          SCALES MATRIX VALUES FROM 0 TO 9      *
00006      C*
00007      C*****
00008      C
00009      DIMENSION LQ(72),L(131),Z(NX ,NY)
00010      DATA NNC,AMAX,AMIN,LOGFL/10.0.,0.,0/
00011      DATA LQ /'0','1','2','3','4','5',
00012      .   '6','7','8','9','A','B',
00013      .   'C','D','E','F','G','H',
00014      .   'J','K','L','M','N','O',
00015      .   'P','Q','R','S','T','U',
00016      .   'W','X','Y','Z','/'/
00017      DATA ISTAR/**/',IMAX/65/
00018      C
00019      C
00020      C
00021      LOGFL = 0
00022      AMIN = 0.0
00023      AMAX = 0.0
00024      NNC = 10
00025      NXMX=NX
00026      C ASPECT = DY/DX
00027      ASPECT = 1.0
00028      JMAX = .6*NY/NX*IMAX*ASPECT + .5
00029      NC = MIN0(NNC,36)
00030      ZMIN = AMIN
00031      ZMAX = AMAX
00032      ANX = NX
00033      ANY = NY
00034      IF (ABS(ZMAX-ZMIN) .GT. 1.0E-5) GO TO 40
00035      ZMAX = Z(1,1)
00036      ZMIN = ZMAX
00037      DO 20 I=1,NX
00038      DO 20 J=1,NY
00039      ZMAX = AMAX1(ZMAX,Z(I,J))
00040      20 ZMIN = AMIN1(ZMIN,Z(I,J))
00041      IF (ABS(ZMAX-ZMIN) .LT. 1.0E-20) RETURN
00042      40 IF (LOGFL .EQ. 0) GO TO 80
00043      IF (ZMIN .GT. 0.) GO TO 60
00044      WRITE (6,220)
00045      RETURN
00046      60 HZMAX = ZMAX
00047      HZMIN = ZMIN
00048      ZMAX = ALOG10(ZMAX)
00049      ZMIN = ALOG10(ZMIN)
00050      80 SC = (NC*2.-1.E-5)/(ZMAX-ZMIN)
00051      DX = ANX/IMAX
00052      DY = ANY/JMAX
00053      Y = ANY+DY
00054      DO 180 JJ=1,JMAX
00055      J = JMAX -JJ +1
00056      Y = Y-DY

```

```

00057      M = Y + .5
00058      DM = AMAX1(Y - M + .5, 1.0E-10)
00059      IF(M.EQ.0) DM = 1.0
00060      IF(M.GE.NY) DM = 0.0
00061      X = 0.
00062      DO 160 I=1,IMAX
00063      X = X+DX
00064      IF (I*N.EQ. 1 .OR. I*N.EQ. IMAX*NMAX) GO TO 140
00065      IF (I .EQ. 1 .AND. N .EQ. NMAX .OR. N .EQ. 1 .AND. I .EQ. IMAX)
00066          GO TO 140
00067      N = X + .5
00068      DN = AMAX1(X - N + .5, 1.0E-10)
00069      IF(N.LE.0) DN = 1.0
00070      IF(N.GE.NX) DN = 0.0
00071      IF (LOGFL .NE. 0) GO TO 100
00072      DMM = AMAX1(1. - DM, 1.0E-10)
00073      DNN = AMAX1(1. - DN, 1.0E-10)
00074      IF(M.EQ.0) DMM = 0.0
00075          IF(M.LE.0) M=1
00076          IF(N.EQ.0) DNN = 0.0
00077          IF(N.LE.0) N=1
00078          NP1=N+1
00079          IF(NP1.GT.NX) NP1=NX
00080          MP1=M+1
00081          IF(MP1.GT.NY) MP1=NY
00082          C = DN*(DM*Z(NP1,MP1)+DMM*Z(NP1,M))+DNN*(DM*Z(N,MP1)+DMM*
00083              .Z(N,M))
00084          GO TO 120
00085 100   C = DN*(DM*ALOG10(Z(N+1,M+1))+(1.-DM)*ALOG10(Z(N+1,M))+((1.-DN)*
00086              .(DM*ALOG10(Z(N,M+1)))+(1.-DMM)*ALOG10(Z(N,M)))
00087 120   IND = 1+SC*(C-ZMIN)
00088          IF (IND .LE. 0) IND = 1
00089          IF (IND .GT. 2*NC) IND = 2*NC
00090          L(I) = LQ(IND)
00091          GO TO 160
00092 140   L(I) = ISTAR
00093 160   CONTINUE
00094
C      INDEX CHANGE TO MAKE PRINTOUT IN MATRIX - LIKE FORM
00095
C
00096
00097 180   WRITE (6,240) (L(I), I = 1 , IMAX)
00098          IF (LOGFL .EQ. 0) GO TO 200
00099          ZMAX = HZMAX
00100          ZMIN = HZMIN
00101 200   WRITE (6,260) ZMIN,ZMAX
00102
C      RETURN
00103 220   FORMAT (/////
00104          115H ****NON-POSITIVE VALUE IN ARRAY TO BE
00105          . CONTOURED LOGARITHMICALLY. PL IS RETURNING WITHOUT PLOTTING )
00106
00107
00108 240   FORMAT (1X,131A1)
00109 260   FORMAT (/17H0MINIMUM VALUE = ,1PE9.3,5X,16HMAXIMUM VALUE = ,E9.3)
00110 280   FORMAT(1H1)
00111  END

```